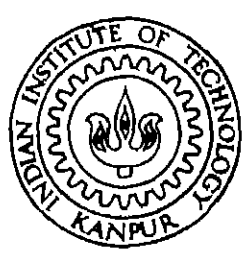
SIMULATION MODEL FOR DYNAMIC SCHEDULING OF BUSES ON A CORRIDOR-A CASE STUDY FOR DELHI RING ROAD

by
SHANTI KUMAR V.V.



DEPARTMENT OF CIVIL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
June, 1997

SIMULATION MODEL FOR DYNAMIC SCHEDULING OF BUSES ON A CORRIDOR - A CASE STUDY FOR DELHI RING ROAD

A Thesis submitted for the partial

fulfilment of the Requirements

for the Degree of

Master of Technology

by

SHANTI KUMAR V.V

DEPARTMENT OF CIVIL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY

KANPUR

JUNE, 1997.

CERTIFICATE

This is to certify that the thesis " SIMULATION MODEL FOR DYNAMIC SCHEDULING OF BUSES ON A CORRIDOR - A CASE STUDY FOR DELHI RING ROAD " submitted by Shanti Kumar V.V in partial fulfilment of the requirements for the degree of Master of Technology of the Indian Institute of Technology, Kanpur, is a bonafide research work carried out by him under my supervison and guidance. The work embodied in this thesis has not been submitted elsewhere for the award of a degree

June 7 th, 1997.

(Dr. B. R. Marwah)
Professor
Dept. Of Civil Engg.
Indian Institute of Technology,
Kanpur.
U.P 208016 INDIA

DEDICATED

to

MY BELOVED PARENTS

ACKNOWLEDGMENTS

I take this opportunity to thank profoundly my thesis supervisor, Dr B.R. Marwah for his encouragement and guidance to complete my thesis work It was indeed a pleasure and privilage to have worked with him

I also thank Dr S P Palani Swamy, and Dr Partha Chakrabothy whose course work had helped me a lot in completing my thesis work

I thank research scholars Mr F A. Kidwai, Mr K H. Reddy and Mr. Bhuvanesh Singh for their valuable advices, and help throughout my cousre work.

I thank Mrs. & Capt R K. Sharma for their affection and support I am greatful to them for providing me a home away from home.

I thank all of my friends, especially Srikant, Raj Gopal, Eswar, Pandu, Giri, Shehu, Shiva and RamP for providing me a nice environment to work here

I also thank my coursemates Ms. Malini, and Mr. Rayi Agrawal, and all my juniors for their cooperation

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

A model is developed to simulate the public transit system on a corridor with Object Oriented Approach This model has the capability for dynamic scheduling of vehicles on a major corridor of a metropolitan city This model simulates the public transit system on any route network, and suggests the optimal utilization of vehicles which minimizes the passenger waiting times at all stops This model has a flexibility with respect to arrival of passengers to a stop A stop may have any one of Exponential, Normal, or Gamma probability distributions of passenger arrival, and different parameters of distribution This model displays the location of buses on the corridor of a city network at every five minute interval using concurrent mode of animation This model supports the interactive graphics which enables the user to obtain the information of stops and buses, or to quit the simulation process at any instant of time. This model is tested with a case study of Dlehi ring road, on which the public transit system is simulated A tentative timetable covering the morning peak period is prepared with 24 buses and is fed to the model The model has suggested 19 additional buses, and revised the timetable.

CHAPTER I

INTRODUCTION

1.1 PROLOGUE

Public transport has become an integral and essential feature of cities in both developed and developing countries In populous and developing countries like India, where urbanization trend is on the increase, mass transit system like commuter rails and bus transportation are mostly unavoidable. In most of the Indian cities, public transportation is inadequate and congestion is severe Despite massive infusion of investment, the transportation crisis in metropolitan areas has been growing Traffic congestion, air pollution, urban sprawl, and the need for energy conservation provide convincing evidence that drastic actions must be taken to improve the mass transit system in India in order to alleviate these problems

In India, among various means of public transportation, the bus transport is most important and dominating because of its door-to-door accessibility and flexibility in operation The success of public bus transport however depends on the efficient operational management looking to the need of the passengers The demand for bus transportation is very high, trip length is short, and it is a cheaper mode because of the subsidy provided by the government in different ways. Bus transportation system in metropolitan cities involves the movement of large number of people between relatively small number of given locations Bus transportation system should provide adequate mobility to various locations to satisfy the demand at all the locations

In a metropolitan area the bus routes are basically designed in two different ways *direction oriented* - where the bus moves along major corridors in certain directions; *destination oriented* - where the routes are designed to satisfy the demand from one node to the other In direction oriented, the *idea* is that the persons board the bus at different nodes on major corridors, and move with high speeds in specific directions. These routes generally have heavy demand and high frequency of operation The destination oriented routes are designed to satisfy the demand among major O-D pairs and provide accessibility to various nodes. The demand at these major nodes will be generally high, however the demand at various intermediate nodes may not be high. Among various issues of direct concern such as route design, bus allocation, bus scheduling, crew scheduling, and maintenance scheduling, the design of bus routes and bus scheduling may be considered as of prime importance.

In the present planning methods, after the route path is designed, the scheduling plans are prepared based on level of service, normally defined in terms of load factor As the daily demand matrix for the public transport is available, the number of bus trips on a particular route are determined for the whole day. The number of buses that operate on particular route in both directions is decided subject to the operational constraints In this way the timetable is prepared for the day where we have minimum headway during morning and evening peak periods, and on other times the headway is generally twice the peak headway

1.2 PROBLEM STATEMENT

Economy in operation, reliable, and adequate level of service to the users should be the primary objectives of any bus transport management It is imperative that the desired level of services to the passengers should be met by the minimum amount of resources. In order to achieve the above objectives it is necessary to investigate the following

- Structuring of the routes in order to meet the demand in an efficient manner on existing transit network.
- Determination of the minimum fleet size for a specified level of service for the system.
- Determination of the schedules with the minimum fleet size determined above to minimize the overall system cost

For the routes along the major corridors, the specific problems which come up are -

- The significant variation of passenger demand from hour to hour
- The desired frequency during the peak period is quite high
- The arrival of passengers at the bus stops follows different probability distributions which is complex to model

For such routes - the load factor becomes very heavy during certain periods, passenger waiting time at stops also increases considerably causing discomfort, inconvenience, etc The demand on these routes is also significantly affected by some other factors such as closing of cinemas, schools, factories, offices, etc

In our present bus system operation, a fixed number of buses are operated as per a pre-determined time table and no consideration is given to the contingencies or bus breakdowns. In transportation and traffic engineering lot of electronic instrumentation is deployed, but a very little information is reported to improve the public bus transportation system

To improve the system of operation, it is desirable to have dynamic scheduling based on actual demand pattern.

1.3 OBJECTIVES OF THE STUDY

The main objectives of the study includes the following tasks related to the scheduling of public bus transportation system on a route network of a city.

1. Generation of passengers at different stops as per different probability distributions.

2 Simulation of the movement of buses from one stop to the other, and scanning of passengers in the bus and at the stop

3 Estimation of statistics for passengers with respect to waiting time

4 Collection of information about the queues at different stops and decide the optimum scheduling of additional buses that may be available

5 Graphic display of movement of buses on route network at a fixed interval of time, and information of stops and buses

1.4 LITERATURE REVIEW

Anderson et al , (1979) have described a flexible simulation model of an urban bus route in peak hour traffic The simulation program has been presented with particular emphasis on the facility for interactive control of the simulation run and for presentation of overall and detailed statistics from the simulation An independent statistical program system based on the mathematical models underlying the simulation model has been used to provide input parameters for the simulation The prime application of their simulation model is to run the program together with an on-line system that collects data from the real bus traffic and presents its status on a display If a difficult situation occurs, the simulation program can be consulted for testing different ways to tackle the problem They have built the model to handle one bus route that may consists of several service variants

Friedman(1976) formulated a mathematical model of a general public transportation network, and suggested an optimized procedure to schedule the buses' departure times. The author has assumed that the network

operates under known deterministic conditions the lines' routes, the stations' locations, the number of passengers and the traffic intensity are given For the sake of lucidity the model, the model first developed under the assumption that the loading as well as unloading times of passengers are zero Afterwards the consequences of the relaxation of this supposition are investigated The performance index of the system is the average waiting time of passengers, and it has to be minimized by a proper choice of decision variables - i e. the buses' departure times - without violating the constraints of prescribed number of buses, and drivers

Friedman (1975) suggested a rapid solution to the apparently simple problem of finding the minimum number of buses that are required to keep up a two way lane with prescribed departure and travel times with the following assumptions

- The buses traverse only the given lane.
- A bus that arrives first at a terminal will be the first to leave it.

No empty runs buses from one stop to the other This procedure for finding the minimum number of buses that are needed to maintain a two way lane was utilized as part of a dynamic programming algorithm which mean to establish the optimal values of the departure times of buses without violating the constraint of having a given number of them.

Dhingra (1983) has developed a simulation model for the scheduling problem of public bus transportation system, which simulates the flow of passengers on all the nodes of the network and the movement of vehicles on all the routes of the network, there by taking into account the interaction effects of

overlapping, crossing, merging, and diverging routes Routes have been designed based on travel demand, and assumed to be independent of schedules. The use of a vehicle over multiple routes has not been covered.

Ceder (1986) provided alternative methods for constructing bus timetables using passenger load data, and attempted to fulfill six major objectives to evaluate alternative timetables in terms of required resources, to improve the correspondence of bus departure times with passenger demand, to provide alternative timetables for the schedulers' use in specific scheduling situations, to permit direct bus frequency changes for possible exceptions (known to the schedulers) which do not rely on passenger demand data, to allow the construction of the timetables with headway smoothing techniques; and to integrate different headway setting, and different timetable construction methods.

Dumas et al., (1990) presented an algorithm that solves the problem of finding the vehicle schedule which minimizes total inconveniences for travel along a fixed path, where service times at nodes are constrained by time windows, and where inconvenience is modeled using convex functions of the service times. It has been shown that the complexity of the algorithm, expressed as a number of unidimensional minimizations, is on the order of the number of nodes for convex inconvenience functions. An extension was proposed which includes linear costs on waiting times, and discrete service time variables

Scheele (1980) formulated a problem to determine the complete travel pattern and to decide which bus frequencies to use on the various lines The problem is formulated as a non-linear programming problem in the form of a compound minimization problem. The assumptions made to simplify this

problem are - the network of streets on which certain bus lines have been set up is given, the total number of buses is given, the total demand for bus transportation is given in the form of the marginal totals of an origin-destination matrix An iterative algorithm to solve this problem is developed

CHAPTER II

SIMULATION MODEL

2.1 *USE OF DIGITAL SIMULATION*

The mass transportation network of a major urban area necessarily involve a large number of logical sequence of decisions in the flow logic The scheduling process being intricate requires the use of digital simulation Digital simulation may be defined as, 'a numerical technique for conducting experiments on a digital computer with certain types of mathematical and logical models that describe the behavior of the complex system (or some part thereof) over extended periods of real time' (Naylor et al, 1967) Simulation serves different functions in system analysis and design Digital simulation has the following additional advantages -

1 Data can be generated which enables the researcher to study and experiment with complex internal interactions of a given system and thus better understanding of the nature of interactions.

2 Simulation of complex systems yield valuable information for identifying the dominant parameters governing the system operation

3 It is useful in validating the component models in conjunction with the global modeling of a system

2.2 *OBJECT ORIENTED PROGRAMMING (OOP) APPROACH*

This model is developed with OOP approach in C^{++} on Turbo C^{++} platform This section gives a brief *idea* of OOP approach, and the use of this approach in this model This section helps in understanding the OOP approach, and thus the program very easily, however, the readers having the sufficient knowledge of OOP approach may skip this section with out any difficulties

OOP is a way of organizing programs The emphasis is on the way programs are designed, not on the details of individual operators. In particular, OOPs are organized around objects which contain both data, and functions the act on that data A class is a template for a number of objects The fundamental *idea* behind OOP approach is to combine both data, and the functions that operate on that data into a single unit, which is called an object An object's functions called member functions in C^{++}, provide the only way to access its data If for example, you want to read a data item in an object, you call a member function in the object It will read the item, and return the value to you The data is hidden, so it is safe from accidental alterations. Data, and its functions are said to be encapsulated into a single entity If you want to modify the data in an object, you know exactly what

functions interact with it the member functions in the object No other function can access the data This simplifies writing, debugging, and maintaining the program

A C^{++} program typically consists of a number of objects which communicate with each other by calling one others member functions Thinking in terms of objects, rather than functions has a surprisingly helpful effect on how easily programs can be designed This results from the close match between objects in the programming sense, and objects in the real world (Lafore, 1994)

2.3 *MODEL DESCRIPTION*

In the previous section the concept of OOP is briefly introduced This section, now describes the model, and discuss how the concept of OOP is used to develop this model

This model covers the simulation of public *Bus* transportation system on a route network of a city given travel demand using heuristic approach The stochastic nature of flow of passengers and vehicles in a network necessitates the use of simulation modeling to achieve the objectives mentioned in section 1 3 The simulation model developed in this study simulates at micro level the arrival of passengers at all *Stop*s as per different probability distributions, and the movement of *Bus*es from one *Stop* to the other. The salient features of the model that accounts various complexities of the system are as follows -

1 The model simulates simultaneously the flow of passengers at all the *Stops* of the network, and the movement of flow of vehicles between all *Stops* of the network

2 The model takes into account the stochastic nature of passenger arrivals For each node, a probability distribution and its parameters are to be specified reflecting the arrival gap characteristics Exponential, Normal, and Gamma distributions have been associated with each *Stop* for the complete simulation run

3 This model suggests the schedule of an extra *Bus* from any of the terminals which may have extra *Buses*, based on some heuristic approaches Also the user can accept or reject the suggestion made by this simulation model

4 This model, for every five minute interval displays graphically the locations of *Buses* on the network, and the information of *Stops* and *Buses* using concurrent mode of animation

5 The graphic display is highly user interactive, that the user can *Stop* the simulation run at any instant of time, and quit the graphic display

The program developed for this model is divided into a number of objects Each *Bus Stop* on the network is considered as one object of class *Stop*. Similarly, each *Bus* is also treated as one object of class *Bus* This program can handle a system with any number of *Stops* and *Buses*. If, for example there are M number of *Stops*, and N number of *Buses*, this program dynamically creates memory for M *Stop* objects, and N *Bus* objects The data structures associated with *Stop* objects, *Bus* objects are Linked Lists, and Queues, to maintain the passengers in each *Stop*, and each *Bus*, and to maintain the queue of departing

Buses at each *Stop* The detailed description of *Stop* objects, and *Bus* objects will be given in later sections

2.4 *MODEL INPUT AND OUTPUT*

This model takes input from four different files -

1 In first input file will have the information of all *Stops* on the route network The first element in input file, *'STOP_XY.IN'* will be *MAX_STOP* , the maximum number of *Stops* under consideration on the network of a city The information of each *Stops* with respect to its sequence number, x and y coordinates, the desired probability distribution of passengers arriving at that *Stop* (*EX* for Exponential, *NR* for Normal, and *GM* for Gamma) and its parameters, and the total number of *Buses* available at that *Stop* have to be stored in this file

2 In the second input file the information of all *Buses* will be stored The first element in the input file *'BUS_INFO.IN'* will be *BUS_CAP*, the maximum capacity of *Bus* The information of the each *Bus* (that are presently being operated in the system) with respect to its sequence number, staring terminal, starting time, and direction of movement on the network (*CW* for clock wise, and *ACW* for anti clock wise) will have to be stored in this file.

3. The input file No 3, *'TIME_TAB.IN'* will have the departure times on all *Buses* at each of the *Stop* on the network of the city The *Bus id* number, and its departure time, following the *id* number of *Stop* will be the right format In this

way the departure times of all *Buses* at each *Stop* will have to be stored in this file

4 The input file No 4, '*LINKTIME IN* 'consists of travel times on all the links of the network The link in terms of starting *Stop*, and ending *Stop* has to be mentioned in the input file, then follows the travel time on that link, and the direction of that link, either CW or ACW

5 The input file No 5, '*OD_MAT IN* ' will hold the OD matrix of size *MAX_STOP* in which each element represents daily demand of passengers from one *Stop* to the other

The output of this simulation model is in the form of graphic display mostly, and can be classified into three categories -

1 At every five minute interval the graphic display of the network of the city, and the locations of *Buses* on the network

2 At every five minute interval the graphic display of *Stop* information with respect to the sequence *id* number of the *Stop*, the number of passengers waiting at that *Stop*, the average waiting time of those passengers who board the last departed *Bus* from that *Stop*, the departure time and sequence number of the next *Bus* from that *Stop*, and the number of spare *Buses* that are available at that *Stop*

3 At every five minute interval the graphic display of *Bus* information with respect to the sequence *id* number of the *Bus*, the no of passengers occupied the *Bus*, the departure time and sequence number of previous *Stop*, and next *Stop*

4 Graphic display of suggestion of additional *Bus* that can be scheduled from any of the terminals if satisfies all the conditions

5 The final output file gives the schedule of all *Buses* including the newly scheduled additional *Buses*, from all *Stops*

2.5 MODEL FORMULATION

This section explains in detail the formulation of the model, and its development As explained in section 2 2, depending on the size of the problem, this model creates the *Stop* objects, and *Bus* objects This section is divided into seven parts for the better understanding

In part one the general flow diagram is illustrated which gives the idea of how the different modules or functions of the model are inter-related In parts two and three of this section the data and member functions associated with each *Stop* object, and *Bus* object are explained In part four the beginning of simulation process is described In part five it is explained how the passengers will be generated at each *Stop* in the model. In part six the techniques of scanning the *Stops*, and *Buses* is discussed In part seven the strategy to find the best *Stop* from which an additional *Bus* could be scheduled is described In part eight the method used in this model to compute the location of *Buses* on the route network is explained The last part i e part nine of this section discusses the termination criterion of the model

2.5.1 GENERAL FLOW DIAGRAM OF MODEL

The rough flow diagram shown below gives a precise idea how the model works, or how the various functions are inter-related to each other However each function and its sub-functions will be explained in later sections with the help of flow charts

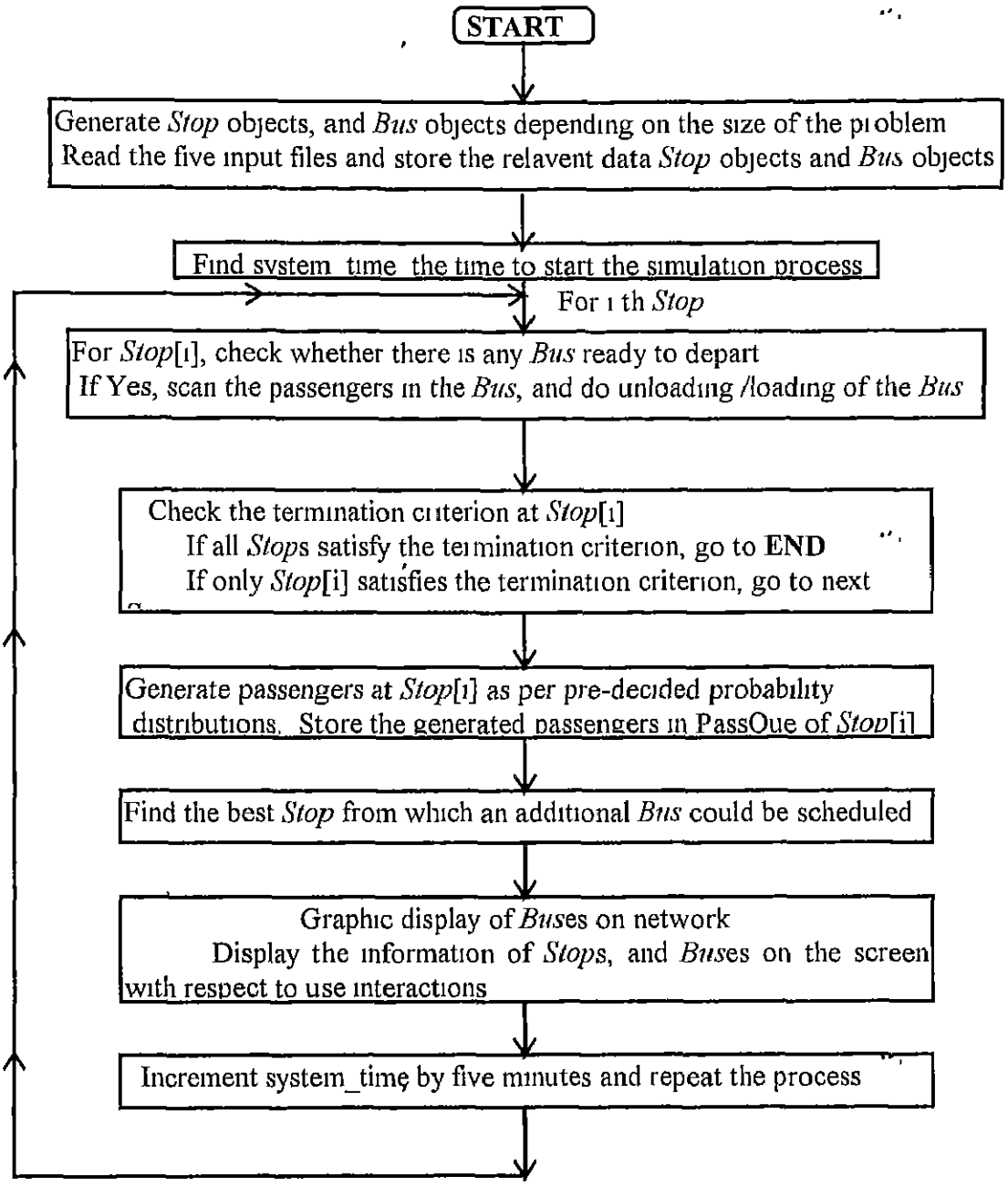


Fig. 2.1 General Flow Diagram of the Model

2.5.2 *STOP OBJECTS*

There will be *MAX_STOP* number of *Stop* objects created by the program Each *Stop* object will have the following data items, and member functions to access the data from outside functions, and to interact with its data

a) **DATA ITEMS** :

- *id* Sequence number of the *Stop*
- *x_co, y_co* X, and Y coordinates of *Stop*
- *dist* Desired probability distribution of passenger arrivals at *Stop*
  Can be any one of *Exponential*, *Normal*, or *Gamma*
- *par1* First parameter of the desired probability distribution
- *par2* , Second parameter of the desired probability distribution
- *tot_bus* . Total number of *Buses* a *Stop* has
  It is equal to zero if a *Stop* is not a terminal
- *free_bus* . Spare or additional *Buses* that are available from a *Stop*
- *tot_wait* . Total waiting time of all passengers served at the *Stop*
- *avg_wait* . Average waiting time of those passengers who board the last *Bus* departed from *Stop* at any instant of time
- *PassQue* The queue of passengers arrived at the *Stop* as per particular distribution Each element of *PassQue* will consists of the arrival time of a passenger, and his/her destination When ever a passenger is generated, he/she will be stored in the *PassQue* of respective *Stop*, and similarly when a *Bus* departs the *Stop* he/she will be deleted from the *PassQue*, and stored in *PassLink* of that *Bus*.

- *BusQue* The queue of departing *Buses* from the *Stop* in the increasing order of their departure times Each element in *BusQue* represents the *Bus id* number, its departure time, and its direction of movement In other way, this is the time table for the *Stop* Whenever a *Bus* departs the *Stop*, the first *Bus* in the *BusQue* will be deleted, so that the next *Bus* will be ready to depart

b) **MEMBER FUNCTIONS** :

- *get_data(stop_no, x_co, y_co, distr, par1, par2, tot_bus)*

  This function stores the data obtained from the input file No 1 into the respective *Stop* object

- *get_timetab(bus_no, bus_time, dirn)*

  This function stores the details of departing *Bus* i e the *Bus* number, its departure time , and its direction of movement in the *BusQue* of object *Stop*

- *return_id()*

  This function returns the *id* number of a *Stop*, and is useful to check the *id* number of a desired *Stop*

- *return_first_bus()*

  This function returns the *id* number of *Bus* which is next to depart from the *Stop*.

- *return_first_time()*

  This function returns the departure time of *Bus* which is next to depart from the *Stop*.

- *return_last_time()* .

This function returns the departure time of last *Bus* from the *Stop* This function is useful to check whether the *Stop* is empty or not

- *move_first_bus()* .

  This function deletes the first *Bus* from *BusQue* when it departs the *Stop*, i e when ever a *Bus* departs a *Stop*, *BusQue* of that *Stop* will be updated

- *add_pass()* ·

  This function stores the randomly generated passenger to the *PassQue* of the *Stop*

- *del_pass()*

  This function deletes the passenger from the *PassQue* of the *Stop* when the passenger boards a departing *Bus*

- *population()*

  This function returns the *population* of passengers, i e the number of passengers in the *PassQue* of the *Stop*.

- *return_tot_wait()*

  this function returns the *tot_wait* of the *Stop*

- *return_avg_wtime()*.

  This function returns the *avg_wait* of those passengers who board the previous departing *Bus* from the *Stop*

- *give_free_bus()*

  This function returns *free_bus*, the number of spare buses that the *Stop* has

- *decrement_free_bus()* .

  This function decrements the *free_bus* of the *Stop*, when an additional *Bus* has been scheduled from the *Stop*

- *return_x_co()*

  This function returns the *x_co* of the *Stop*

- *return_y_co()*

  This function returns the *y_co* of the *Stop*

- *return_no_bus()*

  This function returns the number of *Buses* that are yet to pass through the *Stop*

  This function, when called checks the *BusQue* of a *Stop* and returns the number of *Buses* in *BusQue*. If there are no *Buses*, i e if the last *Bus* has departed, this function returns zero which implies the *Stop* is empty

- *return_par1()*

  This function returns the first parameter of the probability distribution at the *Stop*

- *return_par2()*

  this function returns the second parameter of the probability distribution at the *Stop*

- *check_distr(d) :*

  This function compares the distribution *d* given to this function as argument with the desired distribution at the *Stop*, and returns YES if both are same, and returns NO if not

2.5.3 *BUS OBJECTS*

MAX_BUS, the number of *Bus* objects to be created by the program, depends on the *tot_bus* of each *Stop* The *MAX_BUS* is nothing but the sum of *tot_bus* of all *Stops*. But *FLEET_SIZE*, the number of *Buses* being operated

on the route network, is initially equal to the number of *Bus* objects given in the input file No 2 *FLEET_SIZE* will be incremented by one unit when an additional *Bus* is scheduled The value of *FLEET_SIZE* can not exceed *MAX_BUS*; as no additional *Bus* can be scheduled if there are no *free_bus* at each *Stop* Now let us look into the data items, and member functions of each *Bus* object

a) **DATA ITEMS** :

- *id* Sequence number of a *Bus*
- *term_id* Sequence number of starting terminal of the *Bus*
- *dirn.* Direction of the *Bus* movement on the route network

  CW - Clock Wise or *ACW* - Anti Clock Wise
- *prev_stop* Sequence number of previous *Stop*
- *prev_time* Departure time at previous *Stop*
- *PassList:* The Linked List of passengers who had occupied the *Bus* Each element of *PassList* comprises the destination of a passenger When ever the *Bus* reaches a *Stop*, the *PassList* in the *Bus* will be scanned, and all those passengers whose destination is that *Stop* will be deleted from the *PassList*, and the passengers from *StopQue* will be added to the *PassList* of the *Bus* subjected to the capacity constraint of the *Bus*
- *StopQue:* The queue of *Stops* to which the *Bus* has to travel Each element of *StopQue* comprises the *id* number of a *Stop*, and the departure time of a *Bus* at that *Stop*, in the increasing order of time In other way this is the time table for the *Bus* When ever a *Bus* reaches a *Stop*, the first element will be removed from the *StopQue*

b) MEMBER FUNCTIONS :

- *get_data(* id, *term_*id, *dirn)*

  This function gets the data, the sequence number of a *Bus*, its starting terminal, and its direction of movement from input file No 2 and stores them into the *Bus* object

- *get_timetable(stop_*id, *dep_time)*

  This function stores the data, the sequence number of *Stop*, and departure time at that *Stop* into *StopQue* of the *Bus*

- *return_*id*( ).*

  This function returns the *id* number of the *Bus*, and useful to check the *id* of a *Bus*

- *del_bus_stop( )*

  This function deletes the first *Stop* from the *StopQue* of a *Bus* when the *Bus* departs that *Stop*, and thus updates the *StopQue* of that *Stop*

- *return_stop_*id*( )*

  This function returns *id* of the first *Stop* in *StopQue* of the *Bus*, i e the approaching *Stop* number at which the *Bus* is going to *Stop*

- *return_stop_time( ).*

  This function returns the departure time of the *Bus* at the coming *Stop*

- *add_pass(destination)*

  This function adds one passenger from *PassQue* of a *Stop* to *PassList* of the *Bus*

- *del_pass( ).*

When the *Bus* reaches a *Stop*, this function will be called This function deletes all the passenger from *PassList* of the *Bus* whose destination is that *Stop*

- *population( )*

  This function, when called returns the *population* of *PassList* of the *Bus*, i e the number of passengers occupied the *Bus*

- *store_prev_data( )*

  This function stores the previous *Stop id* number, and the departure time of the *Bus*

- *return_prev_stop( ).*

  This function returns the sequence number of a previous *Stop*

- *return_prev_time( )*

  This function returns the departure time of the *Bus* at the previous *Stop*

- *return_no_Stops( )*

  This function returns the number of *Stops* the *Bus* has to pass through. In other view, this funcion gives the number of *Stops* in *StopQue* of a *Bus*

- *check_dir(d)*

  This function compares the direction of *Bus* and the direction *d* passed to the function as an argument and returns YES if they are same, and NO if not

2.5.4 *BEGINNING THE SIMULATION*

Once the *Stop* and *Bus* objects are created, and the input is given to the model, the model calculates the *system_time*, the time to start the simulation process. In this model, the system time is defined as the time 30 minutes

before the minimum departure time of first *Bus* of all *Stops* Expressing mathematically,

$$system\_time = \text{Min}\ \{\text{dep\_time of } first\_bus \text{ from } Stop[i], \forall i\} - 30 \text{ minutes}$$

Hence, at *system_time*, the model starts simulating the public transit system on a given route network Thereupon, *system_time* will be incremented by 5 minutes During each of this five minute interval, passengers will be generated at all *Stops*, which will be explained in detail in section 2 5 4 each *Stop* and *Bus* will be scanned at every five minute interval and boarding and alighting of passengers will be carried out The section 2 5 5 explains the same process

Also, each *Stop* will be examined during each five minute interval to see whether an additional *Bus* can be scheduled from that *Stop* or not If there are more than one *Stop* from which an additional *Bus* could be scheduled, then that *Stop* which has the maximum saving in waiting time will be chosen to provide the additional *Bus* The evaluation of alternate strategies to find *best_stop* from which an additional *Bus* could be scheduled will be explained in section 2 5 6 At every increment of *system_time*, the locations of all *Buses* on the network will be displayed graphically To display the *Buses* on the network the position of each *Bus* at *system_time* is to be calculated The section 2 5 7 briefly explains the same Accordingly the model will carry through the simulation process until the termination criterion is found, which will be explained in section 2.5.8

2.5.5 GENERATION OF PASSENGERS

This section explains how the passengers are generated randomly at each *Stop* as per specific probability distributions and parameters obtained from input file No 1 This model can handle the passenger arrivals as per Exponential, Normal, and Gamma distributions Each *Stop* can have different, or same probability distributions, and the parameters In input file No 1, the desired probability distributions, and its parameters have to be specified for each *Stop*

The function *generate_pass( )* generates the passengers, assigns them a random destination, and stores them in *PassQue* of respective *Stop* At each five minute increment to *system_time*, this function will be called for each *Stop* if there is at least one departing *Bus* from that *Stop* within 30 minutes from *system_time* In other words, passengers will not be generated at any *Stop* if there is no *Bus* to depart that *Stop* within half an hour from *system_time* This model assumes that passengers start arriving as per pre-decided probability distributions at any *Stop* thirty minutes prior to the departure time of *Bus* from that *Stop* This implies that all passengers are aware of *Bus* timings from each *Stop* Now let us have some idea on how the passengers are generated in this model as per different probability distributions at *Stops* Though the flow charts will be used in later section to understand the logic in a better way

a) **EXPONENTIAL DISTRIBUTION** :

The function *generate_pass( )* generates the passengers at any *Stop* as per Exponential distribution if '**EX** ' is entered in input file No 1 as distribution at that *Stop* Exponentjal distribution will have mean as its parameter. The parameter λ, the mean or inter-arrival rate should be greater than 0, in any

case If the value of λ entered is less than zero, this function takes a 2 0 as the default value of λ , and proceeds with the further calculations In this function inverse-transform technique is used to generate Exponential random variates

Let X have the exponential distribution with mean λ The distribution function is -

$$F(x) = 1 - e^{-x/\lambda} \quad \text{if } x \geq 0,$$
$$= 0 \quad \text{otherwise}$$

So to find F^{-1}, we set $u = F(x)$, and solve for x to obtain

$$F^{-1}(u) = -\lambda \ln(1-u)$$

Thus to generate the desired random variate we first generate a $U \sim U(0,1)$, and then let $x = -\lambda \ln U$ It is possible in this case to use U instead of $1 - U$, since 1 - U and U have the same U(0,1) distribution. This saves a subtraction (Law, and Kelton, 1991).

b) NORMAL DISTRIBUTION :

The function generate_pass( ) generates the passengers as per Normal distribution if NR is the desired distribution at any *Stop* The two parameters of Normal distribution, i e mean μ and variance σ^2 have to be entered in the input file This function however assumes the minimum values of mean and variance as 0 and 1 respectively

The polar method is used in this function to generate N(0,1) random variates (Law and Kelton, 1991) Which is as follows -

1 Generate U_1, and U_2 as IID U(0,1) Let $V_i = 2U_i - 1$ for i = 1,2

Let $W = V_1^2 + V_2^2$.

2. If $W > 1$, go back to step 1.

   Otherwise, let $Y = \sqrt{[(-2\ln W)/W]}$. $X_i = V_i Y$, $i = 1,2$.

   Then X_i, $i = 1,2$ are IID N(0,1) random variates.

Having $X \sim N(0,1)$, we can obtain $X' \sim N(\mu,\sigma^2)$ by setting $X' = \mu + \sigma X$.

c) GAMMA DISTRIBUTION :

The function generate_pass( ) generates passengers as per Gamma random variates if the desired distribution at a *Stop* is entered as GM. The parameters of Gamma distribution are: shape parameter $\alpha > 0$, the scale parameter $\beta > 0$. The mean and variance of Gamma distribution are $\alpha\beta$, and $\alpha\beta^2$ respectively. This functions takes 1, and 1 as the minimum values of α, and β. The algorithm, recommended by Cheng is used in this function to generate Gamma random variates, which generates gamma(α,1) random variates (Law, and Kelton, 1991).

The prespecified constants are $a = 1 / \sqrt{(2\alpha - 1)}$, $b = \alpha - \ln 4$, $q = \alpha + 1/a$; $\theta = 4.5$; and $d = 1 + \ln \theta$.

1. Generate U_1, and U_2 as IID U(0,1).
2. Let $V = a \ln[U_1/(1-U_2)]$, $Y = \alpha e^V$, $Z = U_1^2 U_2$, and $W = b + qV - Y$.
3. If $W + d - \theta Z \geq 0$, return $X = Y$. Otherwise, proceed to step 4.
4. If $W \geq \ln Z$, return $X = Y$. Otherwise, go to Step 1.

Given $X \sim$ Gamma(α, 1) random variate one can obtain $X' \sim$ Gamma(α,β) variate for any $\beta > 0$, from $X' = \beta X$.

ASSIGNING THE DESTINATIONS :

As stated earlier, the function generate_pass( ) generates the passengers as per pre-decided distributions, and assigns them a destination

randomly. The destinations will be assigned to passengers with the help of OD matrix entered in input file No. 5. Each element of OD matrix represents the daily demand from one *Stop* to the other. A function OD_manipulation( ) is used to manipulate the initial OD matrix so that each element of manipulated OD matrix represents the cumulative percent of trips (daily demand) from one *Stop* to the other.

In the function generate_pass( ) the manipulated OD matrix will be used. A random variate U ~ U(0,1) will be generated and if that random number matches with any block of cumulative percent of trips of all destinations for a given *Stop*, that destination will be the passenger's destination. The flow chart will illustrated in section 3.3.4 to explain the same logic.

2.5.6 *SCANNING STOPS AND BUSES*

In this section let us see how this model scans each *Stop*, and each *Bus*, and how it alights/boards the passengers from/to each *Bus*. This process will be repeated for each *Stop*, and for each increment of *system_time*.

For each increment of *system_time*, following steps will be taken For i^{th} *Stop*,

1. Calling *return*_id( ) of *Stop*[i], the *id.* of that *Stop* is stored in *Stop_no*.
2. The passengers waiting at *Stop*[i] is obtained by calling *Stop*[i].*population*( )
3. From the *BusQue* of *Stop*[i] the departure time of *Bus* , and its *id.* are noted as *bus_time*, and *bus_no* respectively. This implies, the *Bus* whose *id.* is *bus_no* departs the *Stop*[i] at *bus_time*. Now the *system_time* is compared with *bus_time*.

4. The *system_time* equal to *bus_time* implies that that *Bus* is about to depart. Now the boarding and alighting of passengers to and from the *Bus* at *Stop*[i] will be done.
5. The *system_time* greater than *bus_time* implies that that *Bus* has already departed the *Stop*. In this case also, the alighting and boarding of passengers from and to the *Bus* at *Stop*[i] will be carried out.
6. The *system_time* less than the *bus_time* implies that the *Bus* has not arrived the *Stop*[i]. In this case, nothing will be done as there is no *Bus* at *Stop*[i]. Now the same process from step-1 will be repeated for next *Stop*, i.e.$(i+1)^{st}$ *Stop*.

BOARDING AND ALIGHTING OF PASSENGERS :

Alighting and boarding of passengers from and to a *Bus* will be done only when the system time is greater than or equal to *bus_time*. The following steps will be taken to achieve this :

1. Scan the passengers in the *PassLink* of the *Bus*.

   If there are any passengers whose destination is equal to *Stop_*no, unload them from *PassLink* of the *Bus* by calling the member function *del_pass*( ) of that *Bus*.
2. Call the member function *population*( ) of the *Bus*, obtain the *population* in that *Bus*, i.e. the number of passengers occupied the *Bus*.
3. Compare the *population* in the *Bus* with *BUS_CAP*, the maximum capacity of the *Bus*. If the *population* is equal to *BUS_CAP*, then the remaining steps will not be executed, as the *Bus* can not accommodate any ,more passengers. In this case, the allow the *Bus* to depart the *Stop*.

4. If the *population* in the *Bus* is less than *BUS_CAP*, load the *Bus* with the passengers from *Stop*[i]. Check the arrival time of first passenger in the *PassQue*.
5. Scan the *population* at the *Stop*, i.e. the passengers waiting at the *Stop*[i] by calling the member function *population*( ) of *Stop*[i].
6. If the *population* at *Stop*[i] is greater than zero, and if the passenger's arrival time is before the departure time of the *Bus*, load the passenger into the *Bus* by calling the member function *add_pass*( ) of that *Bus*. If the arrival time passenger is greater than *bus_time*, allow the *Bus* to depart, as the passengers in the *Stop* arrived after the departure time of the *Bus*. After loading the passenger to the *Bus*, delete that passenger from *PassQue* of *Stop*[i] by calling *del_pass*( ) of *Stop*[i].
7. If the *population* at *Stop*[i] is equal to zero, i.e. if there are no passengers at the *Stop*[i], allow the *Bus* to depart, as there are no passengers waiting at *Stop*[i].

MOVING THE BUS FROM STOP :

If there are no passengers at the *Stop*[i], or if there is no space in the *Bus* for passengers, or if the alighting and boarding of passengers is finished, then allow the *Bus* to depart the *Stop*. Call *move_first_bus*( ) of *Stop*[i], which deletes that *Bus* from the *BusQue* of *Stop*[i], and updates the *BusQue*.

Once the *Bus* is moved out of *Stop*[i], then the same procedure will be repeated for next *Stop*, i.e. *Stop*[i + 1]. If this procedure is repeated for all *Stop*s, then the next step to be under taken is finding the *best_stop* from which an additional *Bus* could be scheduled. This method is described in the following section.

2.5.7 *FINDING THE BEST STRATEGY TO PROVIDE AN ADDITIONAL BUS*

This section explains the evaluation criterion to find the best strategy to provide an extra *Bus* in to the route network. This evaluation will be carried out at each five minute interval of *system_time*. The function *find_best_stop( )* will be called each time, which examines all the *Stop*s to find the *best_stop* which can provide the maximum saving in waiting time for the passengers.

SELECTING THE ALTERNATIVES :

First the alternative *Stop*s for *best_stop* will be traced out based on some heuristic conditions, then the *best_stop*, the best of all alternatives will be found, and an additional *Bus* will be suggested by the model. The three conditions that have to be satisfied for a *Stop* to be the one of alternative are -

- The basic condition is that the *Stop* must have at least one extra *Bus*. A *Stop* with out an extra *Bus* can not be considered as an alternative as it is not possible to schedule any *Bus* from that *Stop*.
- The condition favoring system operators is that there must be at least some minimum number of passengers waiting at a *Stop* which satisfies the first condition. This condition is imposed keeping in view the operators' benefits. If there are a few number of people waiting at the *Stop* will not make any good for the operators. Any number which suits the system can be considered as the minimum number of passengers. However, in this model, 20 is considered as the

minimum number. More clearly, only that *Stop* which has minimum 20 number of passengers waiting at *system_time* is said to be satisfy the second condition.

- The third condition is that there must be no *Bus* at a *Stop* with departure time lesser than or equal to *system_time* plus some minimum time period. If a *Stop* satisfies the first two conditions, and only if there is no *Bus* from that *Stop* to depart within some minimum period of time from *system_time*, that *Stop* can be considered as the alternative. This minimum time period can be any thing from 0 minute, however minimum time period considered in this model is 5 minute. So, a *Stop* to be one of the alternatives have to satisfy the first two conditions, and there must be no *Bus* leaving that *Stop* with departure time falling in the five minute period from *system_time* onwards.

If there are no *Stop*s which satisfy all the three conditions, then there would be no alternatives, and no additional *Bus* will be suggested. If there is only one *Stop* satisfying all the three conditions, then no evaluation of alternatives is necessary, as there is only one alternative, and that *Stop* will be considered as the *best_stop*, and an additional *Bus* will be suggested from that *Stop*. Only if there are more than one *Stop* satisfying all the three conditions, then evaluation of alternatives will be made to find the *best_stop* as explained here.

EVALUATING THE BEST ALTERNATIVE :

Evaluation of alternatives will be made only if there are more than one *Stop* which is considered as an alternative, i.e. which satisfy all three conditions stated above. For each alternative, first the saving in waiting time will be calculated. Then the alternative which has the maximum saving in waiting time will

be considered as the *best_stop*. The following steps will be followed to calculate the saving in waiting time for each alternative, say for *Stop*[i] -

1. Get *bus_time*, the departure time of next *Bus* from *Stop*[i] by calling *return_first_time*( ) of *Stop*[i]. Calculate the *time_diff*, the difference between the *system_time*, and *bus_time*.
2. *time_diff* = *system_time* - *bus_time*.

   This implies that if an additional *Bus* is scheduled from *Stop*[i], it will save the waiting time of each passenger by *time_diff* minutes.
3. Now get the *population* of *Stop*[i] from *population*( ) of that *Stop*.
4. Saving in waiting time at *Stop*[i] is the product of *population* at *Stop*[i] and *time_diff*.
5. Similarly get the saving in waiting time at all the other *Stop*s if the *Bus* from *Stop*[i] is scheduled at *system_time*. Saving in waiting time at each of the *Stop*s, other than *Stop*[i] is the computed as the product of *population* of that *Stop*, and *time_diff*.
6. Now calculate the total saving in waiting time which is the sum of saving in waiting time of all *Stop*s.
7. In case of movement of *Bus*es on both directions, calculate the total saving in waiting time at a *Stop* in both directions separately, and the direction in which total saving is maximum should be taken into account.

Once all the *Stop*s are evaluated with respect to the total saving in waiting time, the *best_stop* will be that *Stop* with maximum total saving in waiting time. This model evaluate the *best_stop* considering the both directions of movement. Thus and so the function find_*best_stop*( ) finds the *best_stop*, and the

direction of *Bus* to be scheduled from that *Stop*. The model displays a message graphically which states that an additional *Bus* is scheduled from the *best_stop* at the *system_time*, and gives the user the choice to accept it or reject. If the user accepts the suggestion, then the new *Bus* will join the fleet on the network. In this case the *FLEET_SIZE* will be incremented by one, and the *free_bus* of the *best_stop* will be decremented by one, and the simulation continues further. If the user rejects the suggestion made by the model, then the new *Bus* will not be scheduled and the simulation continues further.

2.5.8 COMPUTING BUS LOCATIONS

This section explains how the at every increment of *system_time* the *Bus* positions are calculated to display their locations on the network. The function *get_bus_locations*( ) will be called after the function *find_best_stop*( ). The following steps will be followed by the function *get_bus_locations( )*.

At any *system_time*, for every *Bus* -

1. For a *Bus*, from *StopQue* get the next *Stop* number, and departure time at that *Stop* using functions *return_stop*_id( ), and *return_stop_time*( ).
2. Get the previous *Stop* details for a *Bus* from *return_prev_stop*( ), and *return_prev_time*( ).
3. The *system_time* equal to *prev_time* -1, or *prev_time* implies that the *Bus* is at *prev_stop* at *system_time*. Thus the position of that *Bus* is that of the *prev_stop*. Now plot the position of *Bus* on the network. And go to step 1, repeat for next *Bus*.

4. The *system_time* equal to *next_time* implies that the *Bus* is at *next_stop* at *system_time*. So the position of the *Bus* is that of *next_stop*. Plot the position of the *Bus* on the network, and go to step 1 and repeat for next *Bus*.
5. The *system_time* greater than *prev_time*, and less than *next_time* implies that the *Bus* is in between the two *Stop*s. In this case it is necessary to do some calculations to get the position of the *Bus*. Proceed to step 6.
6. Compute the *time_lag*, the difference of *prev_time*, and *system_time*. Compute the *tot_dist*, the distance between *next_stop*, and *prev_stop* in terms of pixels. Obtain the travel time from *prev_stop* to *next_stop* from input file No. 4.
7. Now calculate the desired speed of the *Bus*, which is equal to *tot_dist* divided by the travel time. Get *move_dist*, distance (in terms of pixels) to move the *Bus* from *prev_stop* is equal to the desired speed of the *Bus* multiplied by the *time_lag*.
8. Calculate the slope of *Bus* movement from *prev_stop* to *next_stop* depending on the direction of the *Bus*. Obtain the angle θ, which is inverse tangent of slope.
9. Now, at *system_time* the current position of the *Bus* on the network is obtained from -

$$Bus.x\_co = prev\_stop.x\_co + move\_dist \times \cos\theta'$$

$$Bus.y\_co = prev\_stop.y\_co + move\_dist \times \sin\theta'$$

where $\theta' = \theta$ if the slope is south east, or north west.
$= 90 - \theta$ if the slope is south west, or north east

10. Plot the *Bus* position on the network. Go to step 1, and repeat for the next *Bus*, until all *Bus*es are plotted.

2.5.9 *TERMINATION CRITERION*

The model will proceed with the simulation till the termination criterion is found. If the last *Bus* from a *Stop* has departed, then that *Stop* is said to be empty. So if all *Stop*s are found empty then the termination criterion is supposed to be satisfied. To check whether a *Stop* is empty or not call the member function return_no_*Bus*( ), which returns the number of *Bus*es in *BusQue* of that *Stop*. If the termination criterion is found, the model displays a message graphically that all *Stop*s are empty, and asks the user to quit the model as no further simulation is possible.

CHAPTER III

PROGRAM STRUCTURE

3.1 ABOUT TURBO C++

As stated in section 2.2 this program is developed with OOP approach using C++ and run on Turbo C++ compiler Version 2.0 in an MS-DOS environment. Turbo C++ provides a powerful environment called *integrated development environment* (IDE) for creating and executing a program. The IDE is completely menu-driven and allows the user to create, edit, compile, and run programs using what are known as *dialogue boxes*. Turbo C++ requires MS-DOS 2.0 or higher, and runs on PC XTs, ATs, PC/2S, and compatibles. It needs at least 640 KB of memory, a hard disk, and a floppy disk drive. The hard disk should have some what more than 6 MB of free space. Turbo C++ will also run in Microsoft Window DOS box. Turbo C++ works with any 80 column monitor either character based (which usually means monochrome) or graphic-based (usually color). Turbo

C^{++} can also accommodate a mouse, how ever it is not necessary; one can perform all operations from the key board.

3.2 MAIN PROGRAM

The main program is divided into 6 major functions, a few of which are further divided into sub functions. This section explains the complete program with the help of flow charts. The six functions which will be explained in later sections are -

1. *graphics_display( )*
2. *create_stops( )*
3. *create_buses( )*
4. *assign_time_table( )*
5. *od_manipulation( )*
6. *do_operation( )*

3.2.1 GRAPHICS_DISPLAY( )

The main program of this model calls five functions stated above. First function to be called by the main program is *graphics_display( ).* This function in itself calls *initgraph( )*, a library function which must be executed before any other graphic-mode function used. This library function switches the computer's display system into appropriate graphics mode. This function requires the *ghaphics.h* header file. After initiating graphics, *graphics_display( )* function calls other graphic functions used in the program to create windows, push-buttons, and text. The steps which will be executed in this function are given below.

1. initgraph( )

2. display_main_window( )
3. display_time_window( )
4. display_push_buttons( )
5. display_text( )

3.2.2 *CREATE_STOPS( )*

This is the function that will be called after the *graphics_display( )*. This function reads the first element *MAX_STOP* of input file No. 1, and creates the *MAX_STOP* number of *Stop* objects, and then stores the data from input file No. 1 by calling the member function *get_data( )* of each *Stop*. This function calculates *MAX_BUS*, the maximum number of Buses that can be used in the system which is the sum of *tot_bus* of all *Stops*.

3.2.3 *CREATE_BUSES( )*

This function opens the input file No. 2 and first reads *BUS_CAP*, the maximum capacity of a Bus. Then it creates the *MAX_BUS* number of Bus objects. Then it reads the input file No. 2, and computes *FLEET_SIZE*, the number of buses being operated initially, which is the number of Buses defined in the input file No. 2. Then it calls *get_data*( ) of each Bus, and stores the data into the respective Bus object.

3.2.4 *ASSIGN_TIMETABLE( )*

Once the required number of *Stop* objects, and Bus objects are created, the next step is to assign the time table to each *Stop* object, and *Bus* object. This file opens the input file No. 3 and reads the time table for each *Stop* of the ring road network, and stores them in respective *BusQue* of *Stops*, and *StopQue* of Buses

by calling the *get_timetab( )* function of each *Stop*, and each *Bus*. The flow chart below explains the same process.

3.2.5 OD_MANIPULATION( )

This function opens the input file No. 4, and reads the OD matrix of size *MAX_STOP*. The each element of OD matrix in the input file represents the daily demand from one *Stop* to the other. This function manipulates the OD matrix using normalization technique in such a way that each element represent the cumulative percentage of daily demand from one *Stop* to the other. The manipulated OD matrix will be used to assign the destinations randomly to the passengers by the function *generate_pass( )*.

3.2.6 DO_OPERATION( )

The functions explained so far in the previous sections open and read the four input files, create the *Stop* and Bus objects, assign the time table to them, and manipulate the OD matrix. But, they do no simulation, and scheduling. The function *do_operation( )* is the important and most critical function, and it does the simulation and scheduling part of the modeling. This function contains sub functions, a few of which contain another level of functions. The sub functions used in this function are -

1. *find_system_time( )*
2. *generate_pass( )*
3. *get_bus_locations( )*
4. *find_best_stop( )*

The function *do_operation( )* will be explained with the help of flow charts in the following sections. Each of the sub functions will also be described.

This function *do_operation*( ) first calls the function find_*system_time*( ) to get *system_time*, the time to start the simulation process. The *system_time*, in this model is defined as the time 30 minutes before the minimum of first bus departure time from all *Stops*. Now let us try to study the functions used in *od_manipulation*( ). They are *generate_pass*( ), *find_best_stop*( ), and *get_bus_locations*( ).

A) ***generate_pass(stop_no)***

This function takes *Stop*_no, the *id* number of a *Stop* at which passengers are to be generated, and generates the passengers as per pre-decided probability distribution for a period of five minutes starting from *system_time*, assigns the random destinations to he passengers, and stores them in *PassQue* of respective *Stop*.. The description of this function is given in section 2.5.4, which gives a precise over view of the function. However in this section let us study the same with the help of flow chart. The figure 3.3 shows the flow chart of the function generate_pass( ).

B) find_best_stop( )

The function find_*best_stop*( ) will be called at each increment of system time by five minutes, which evaluates the best strategy to provide an additional Bus on to the ring road. This function is explained in section 2.5.6 in detail. In this section the flow chart is given to understand the same more clearly. The figure 3.4 shows the flow chart of the function *find_best_stop*( ).

C) ***get_bus_locations( )***

This function at each increment of system time calculates the positions of Buses and plots the Buses on the ring road. The section 2.5.7 explains the function. The flow chart is presented in this section to enable the readers understand the logic used in this function

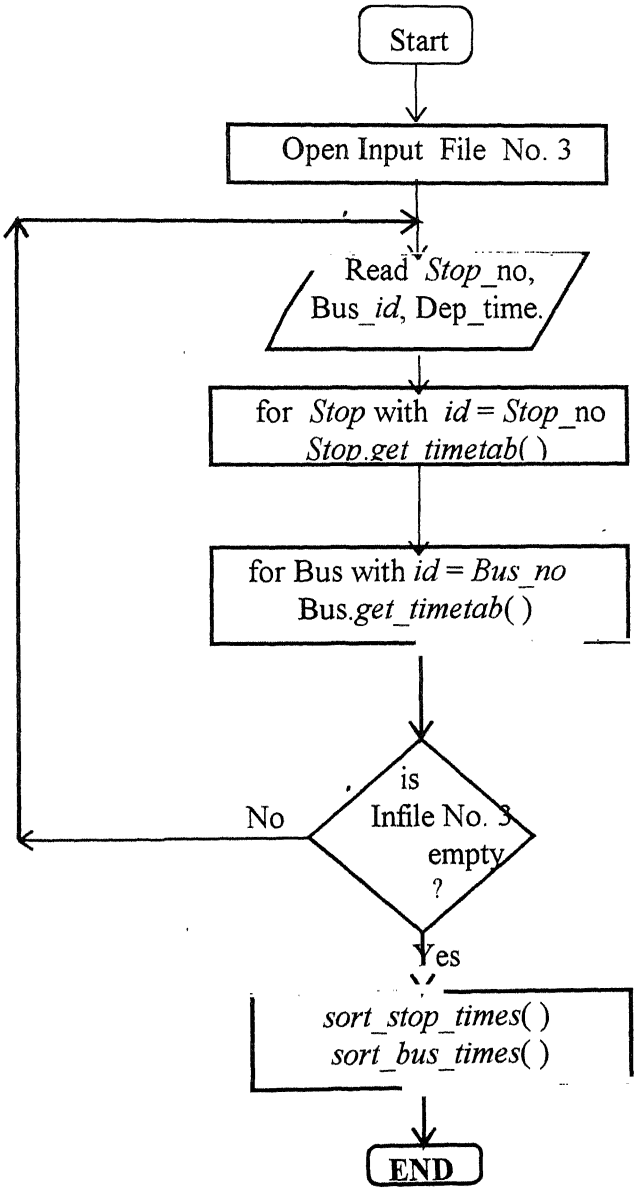


Fig. 3.1 Flow Chart for Function *assign_timetable( )*

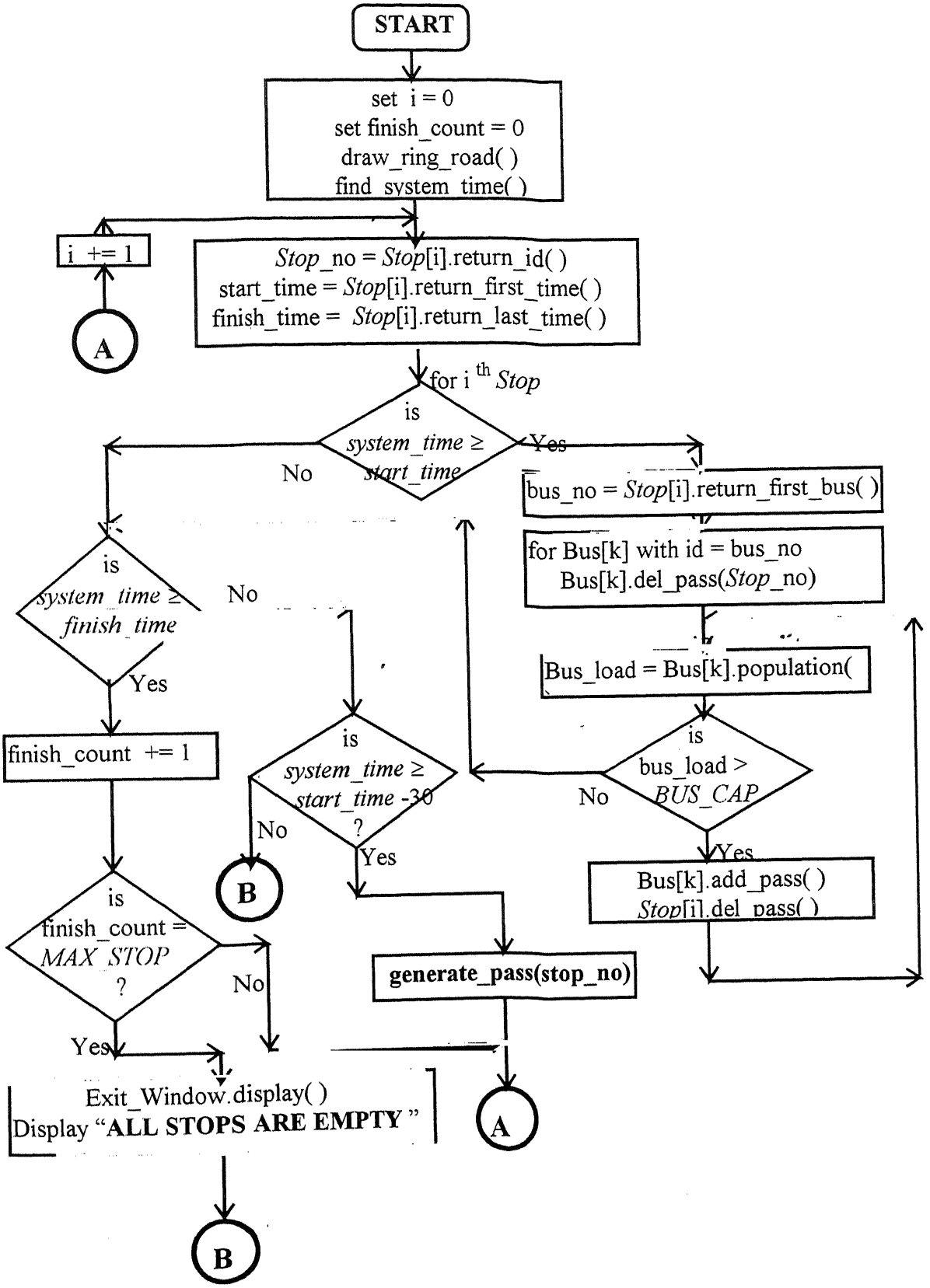


Fig. 3.2a Flow Chart for Function *do_operation( )* (...continued)

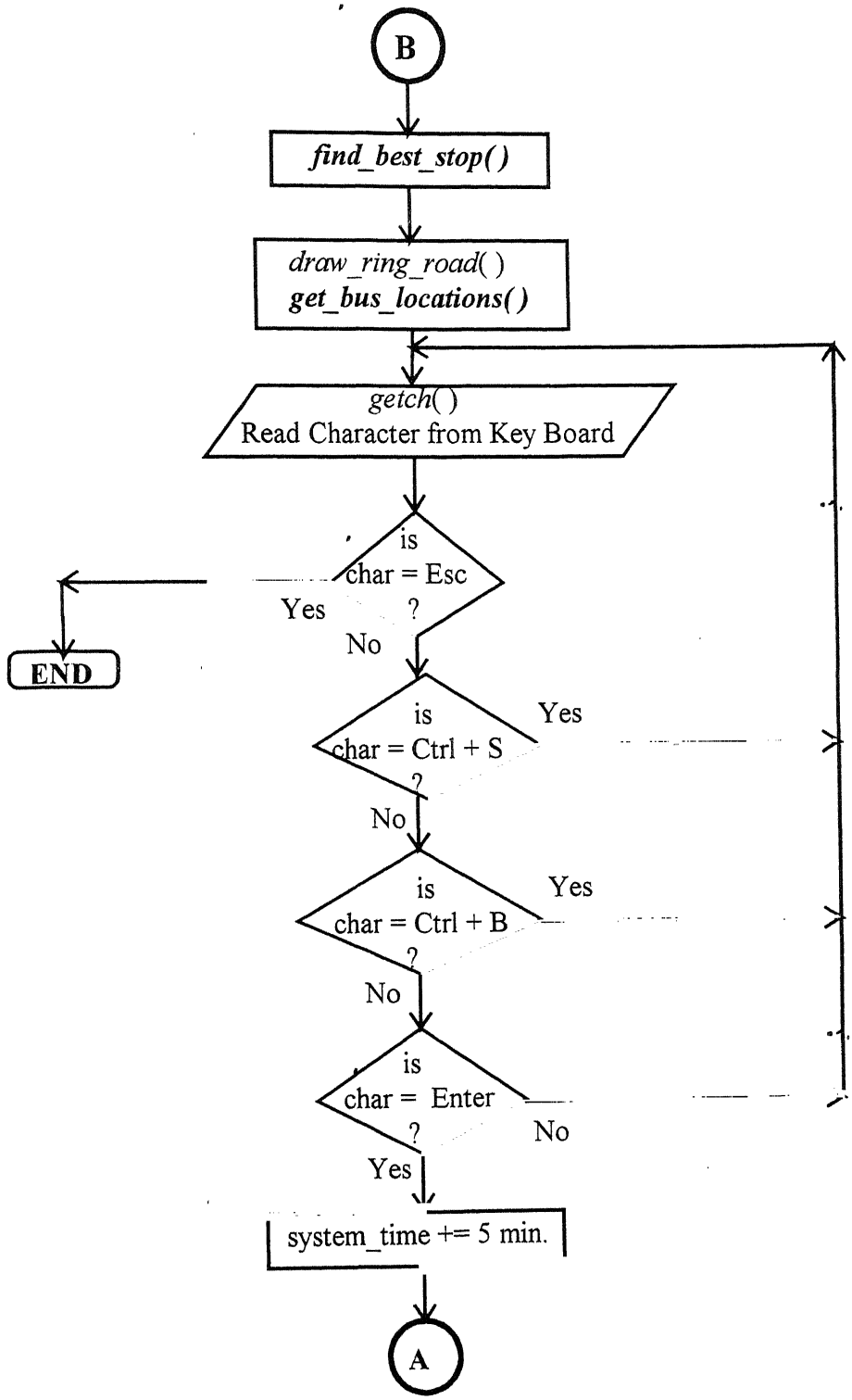
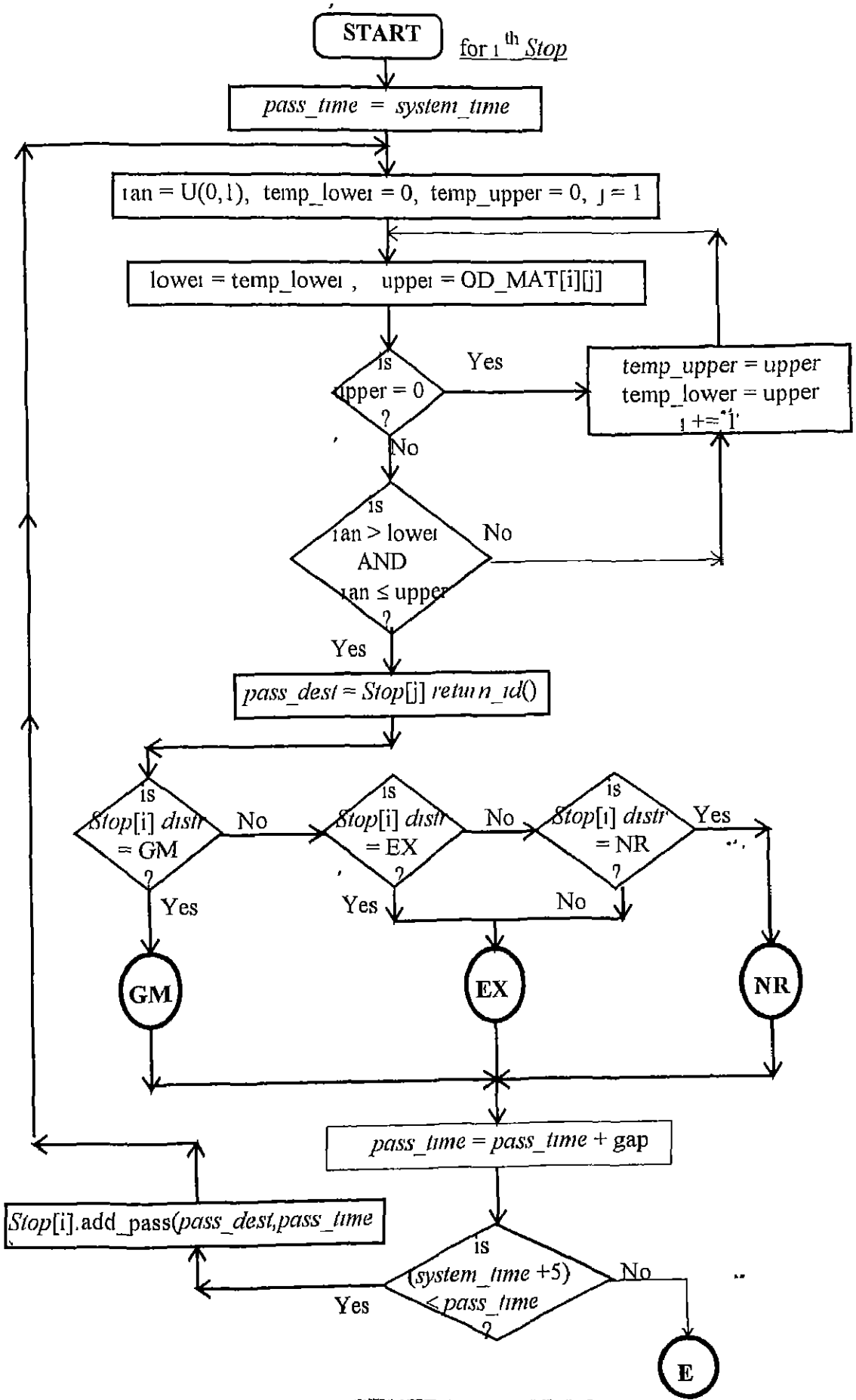


Fig. 3.2 Flow Chart for Function *do_operation( )*



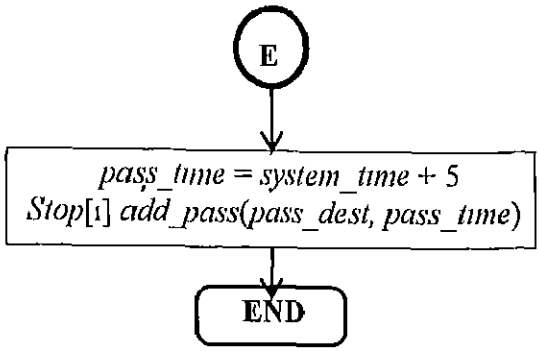


Fig. 3.3 Flow Chart for Function *generate_pass( )*

GM, EX, and **NR** in Fig 3 3 connects to the functions to generate Gamma (Fig 3 3a), Exponential (Fig 3 3b), and Normal (Fig 3 3c) random variates respectively

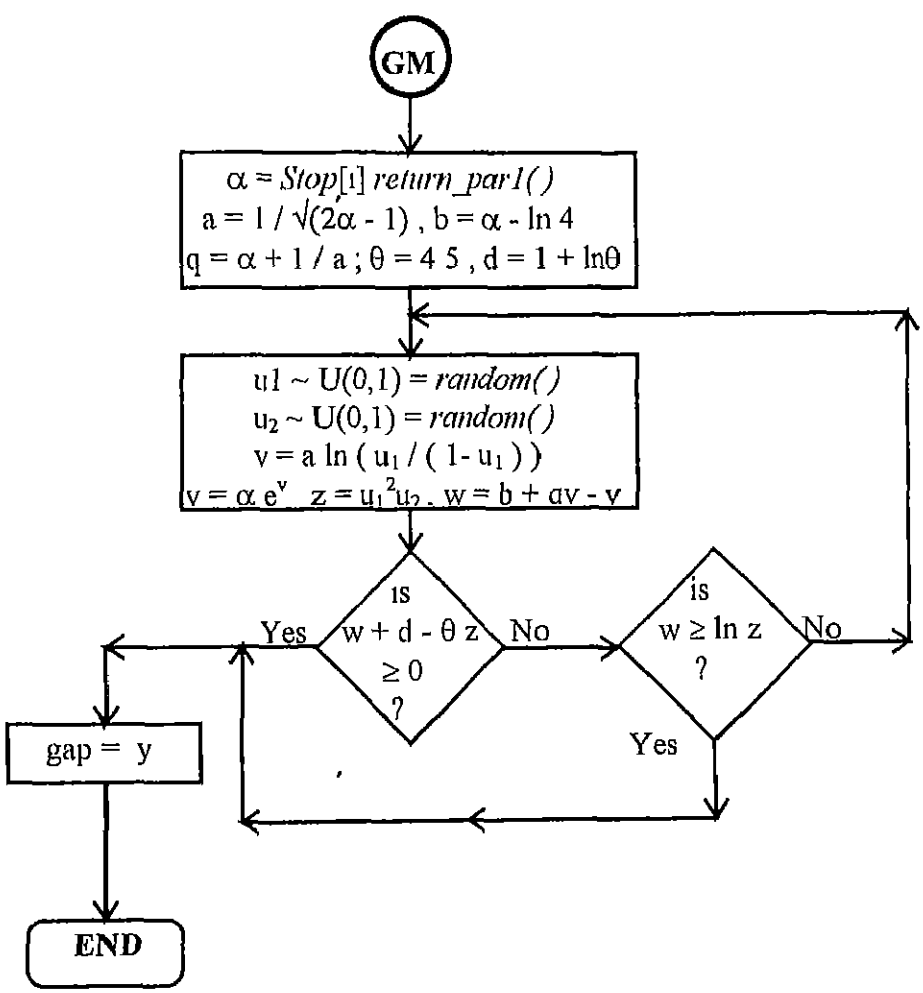


Fig 3.3a Function to generate *Gamma* random variates

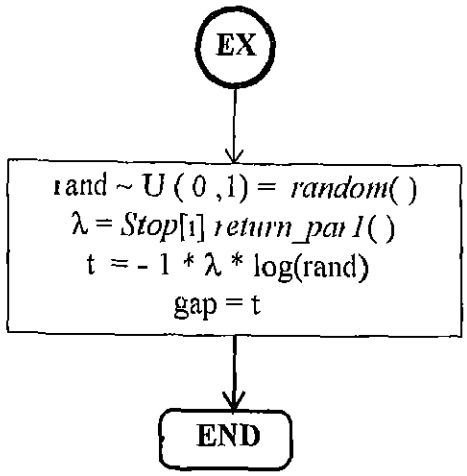


Fig. 3.3b Function to generate *Exponential* random variates

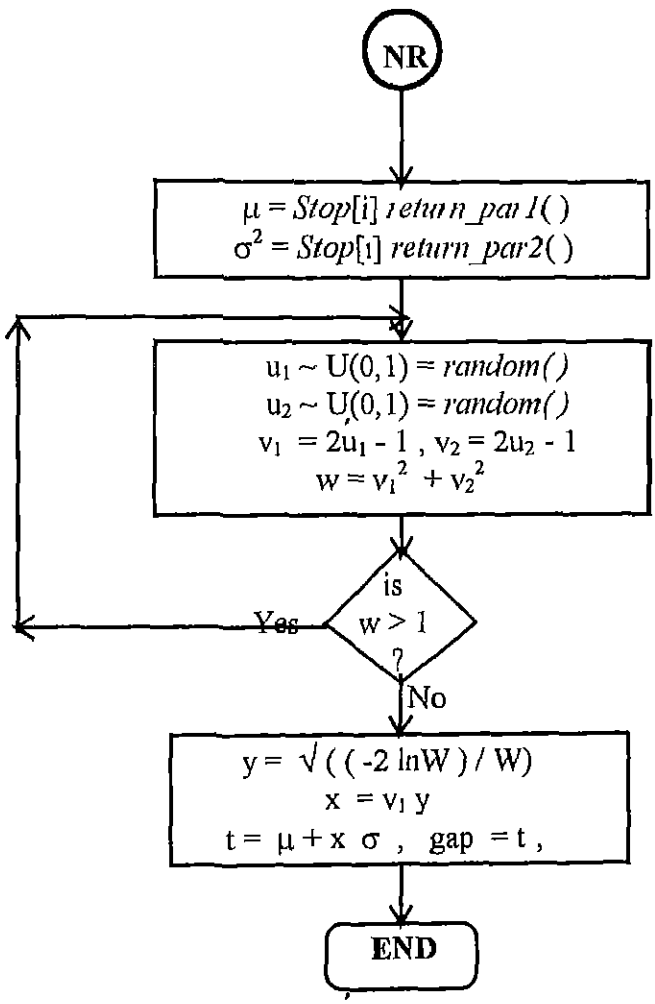


Fig. 3.3c Function to generate *Normal* random variates

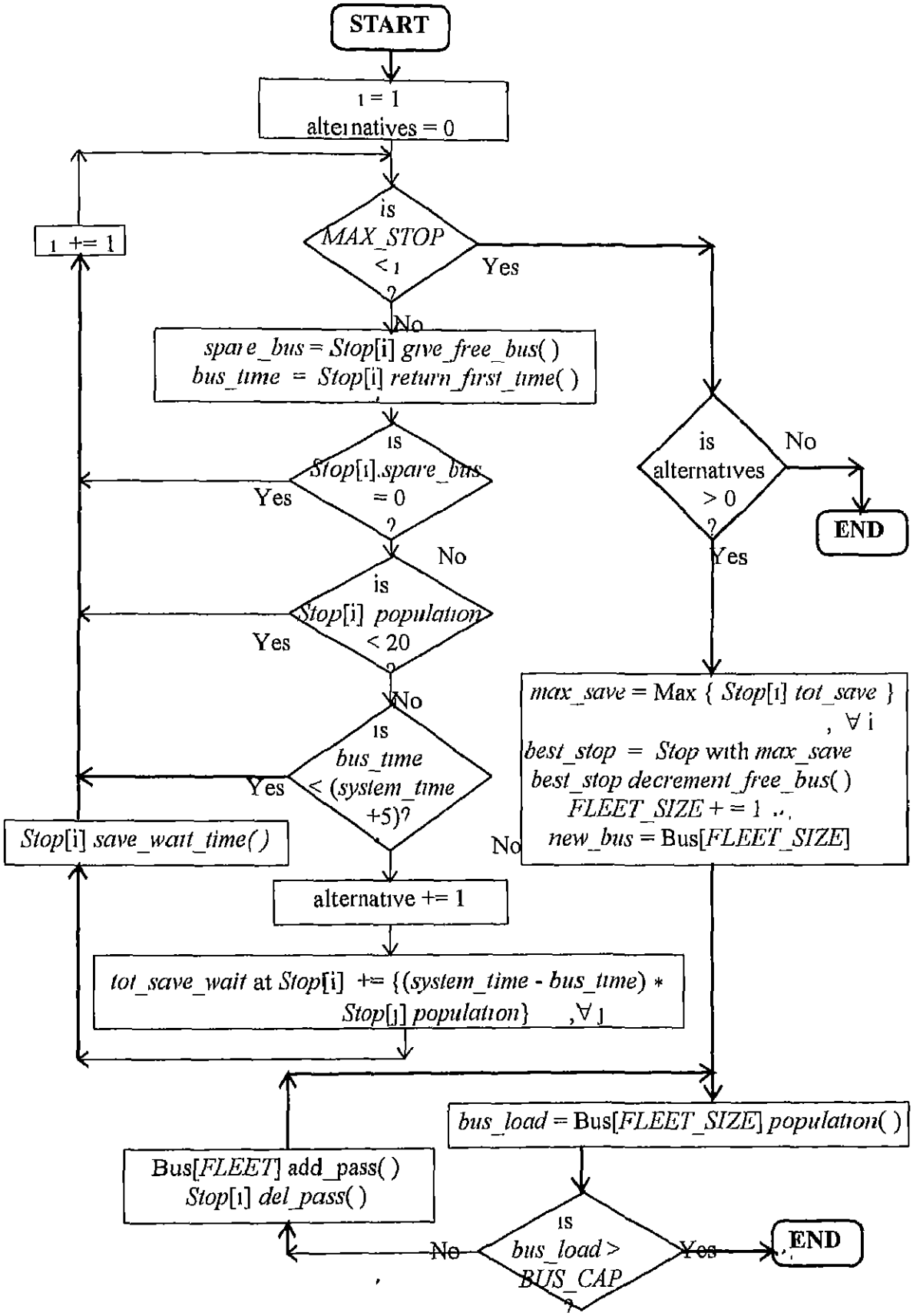


fig 3.4 Flow Chart for Function *find_best_stop( )*

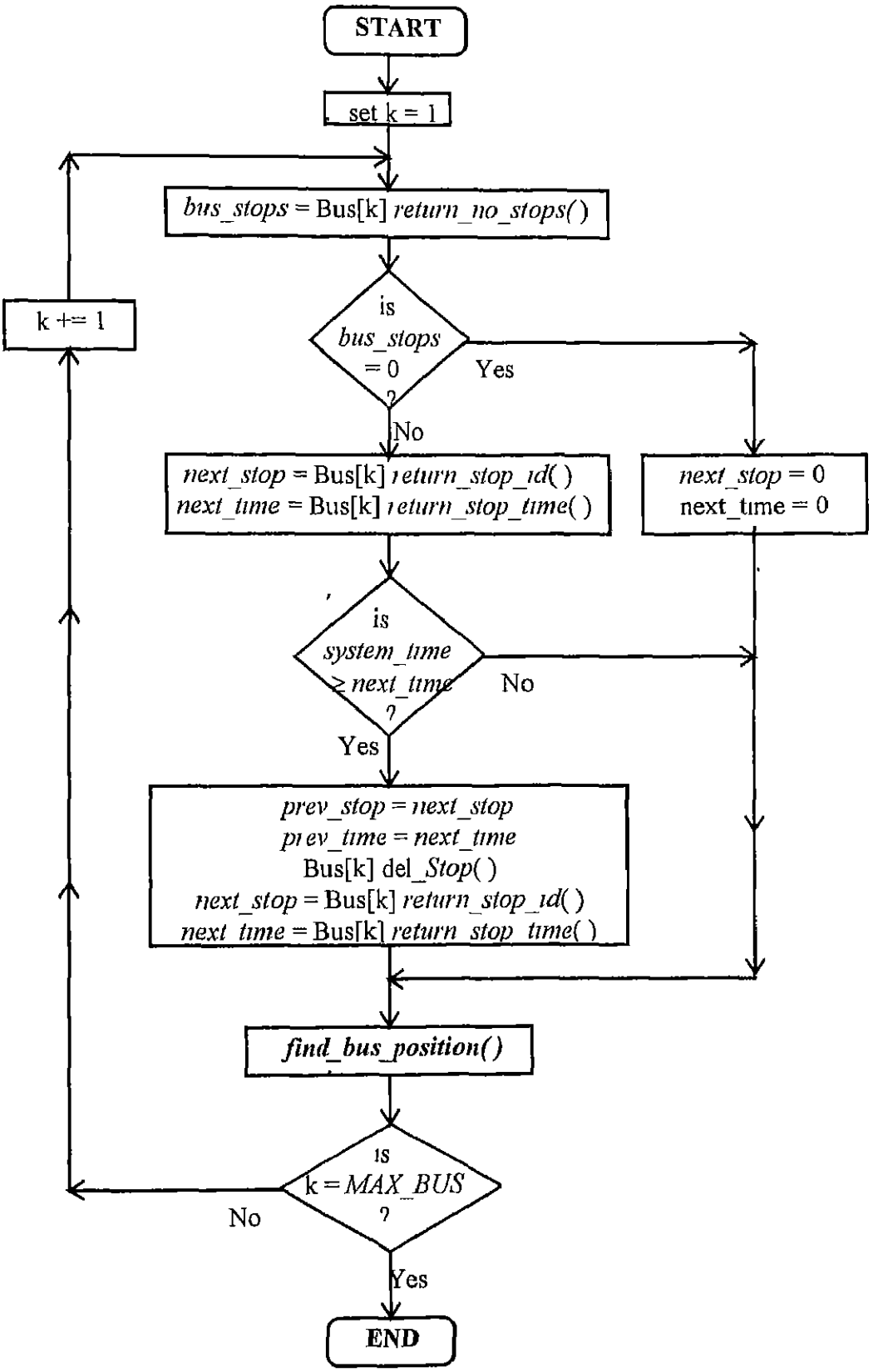
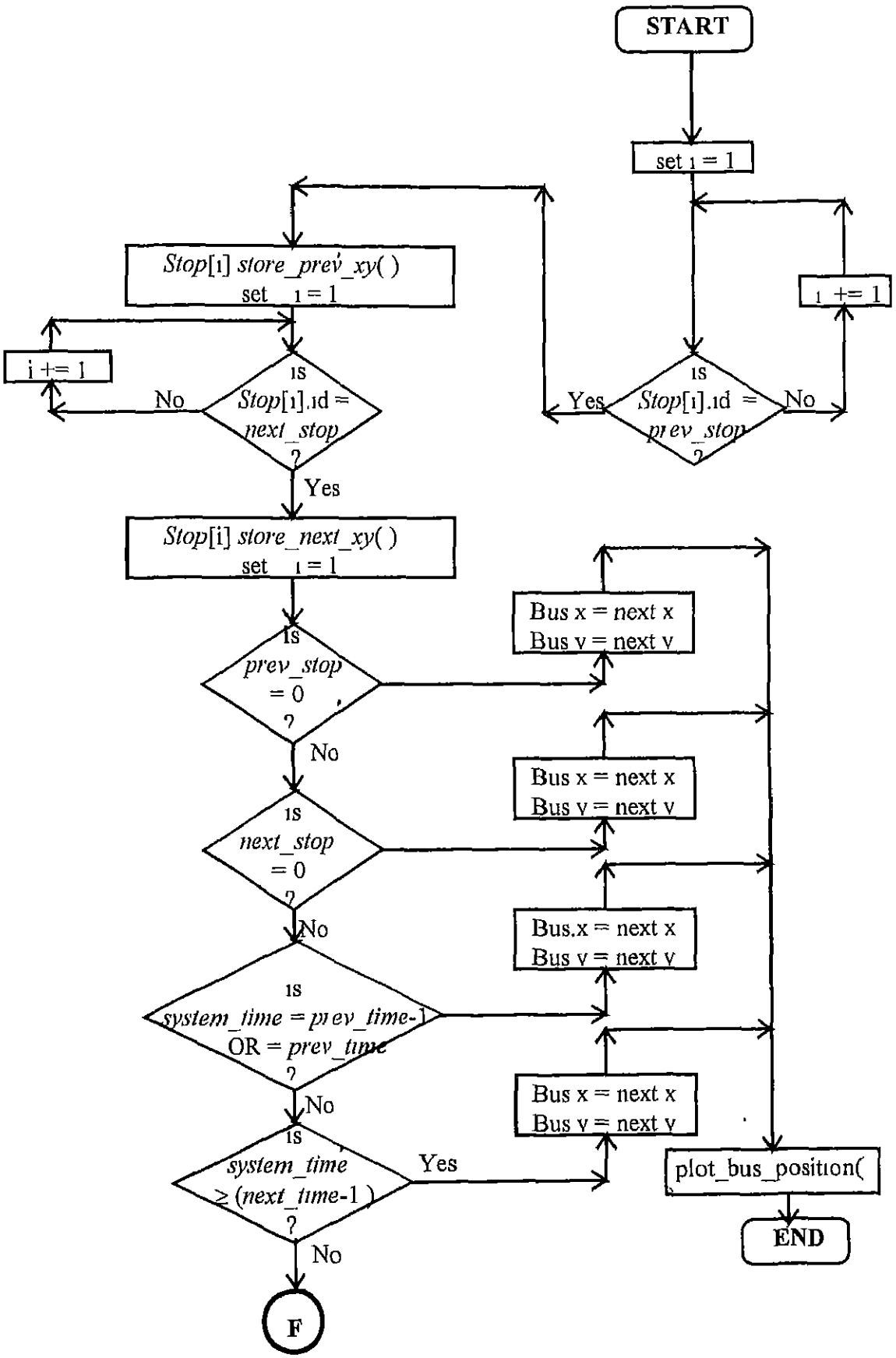


Fig. 3.5 Flow Chart for Function *get_bus_location( )*


START
set i = 1
Stop[i] store_prev_xy( )
set i = 1
i += 1
No
Is Stop[i].id = next_stop ?
Yes
Stop[i] store_next_xy( )
set i = 1
Is prev_stop = 0 ?
No
Is next_stop = 0 ?
No
Is system_time = prev_time-1 OR = prev_time ?
No
Is system_time ≥ (next_time-1) ?
Yes
No
F
Is Stop[i].id = prev_stop ?
Yes
No
i += 1
Bus x = next x
Bus y = next y
Bus x = next x
Bus y = next y
Bus.x = next x
Bus y = next y
Bus x = next x
Bus y = next y
plot_bus_position(
END

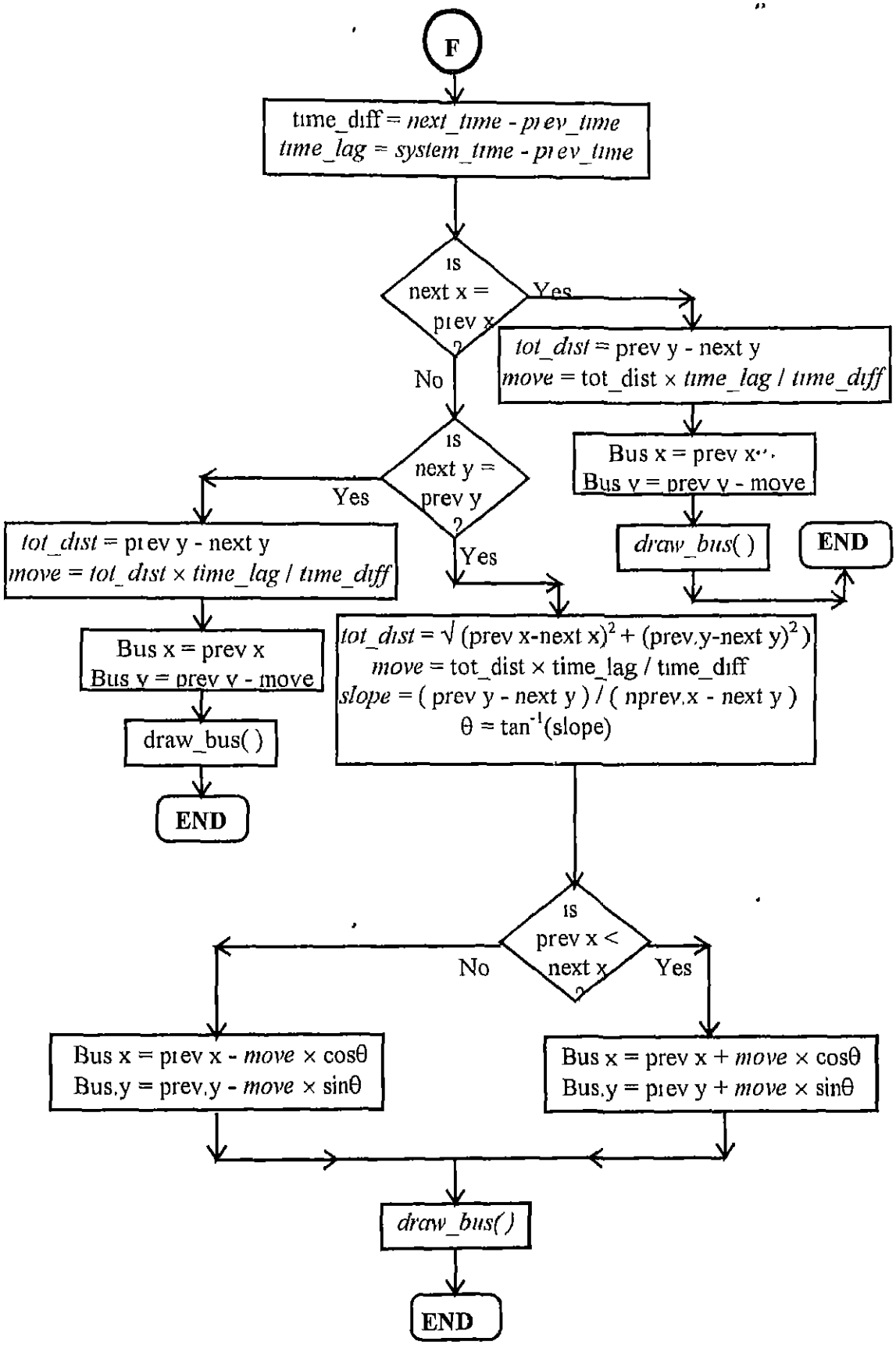


Fig. 3.6 Flow Chart for Function *find_bus_poisition( )*

CHAPTER IV

MODEL APPLICATION

- A CASE STUDY

4.1 INTRODUCTION

The model discussed in the previous sections is tested for a case study of New Delhi metropolitan area The ring road network of New Delhi city is considered and the transit system on the ring road is simulated using this model The input for this model is extracted from the data collected from primary surveys, and secondary sources for the study 'Optimal Design of Urban Bus system' conducted by RITES for New Delhi metropolitan area (RITES, 1993). The data is available for the complete transportation network of New Delhi metropolitan area in terms of link distances, daily demand from one node to the other for all 1200 nodes. The input thus extracted is fed to the model and the output is obtained. The following sections explain the extraction of input and generation of output both in graphics part and non-graphics part.

4.2 INPUT TO THE MODEL

As explained before, the input to the model is extracted from the data collected for the study ' Optimal Design of Urban Bus System ' conducted for New Delhi metropolitan area (RITES, 1993). This section explains how the required input is derived from the complete data First all the bus stops on the ring road network are identified, the daily demand matrix for those bus stops on the ring road is extracted from that of the complete transportation network, the major stops are identified as terminals on the ring road; the travel times on all the links of ring road network are obtained, and a tentative time table for the bus system on the ring road is generated

4.2.1 IDENTIFICATION OF STOPS ON THE RING ROAD :

First the ring road is identified from the map of the complete network of Delhi area, and then all the Stops on the ring road are noted. For Delhi network with 1200 nodes the total number of nodes on the ring road are found to be 76. These 76 nodes are very closely spaced and it would be complicated to display the movement of buses graphically on the screen with such closely spaced nodes Hence some closely spaced nodes are merged in to a single node, thus obtaining 31 nodes on the ring road network. These 31 nodes are identified based on the link distances, and the daily demand from one node to the other. The model is then used to simulate the public transit system on the ring road network of Delhi city with 31 bus stops on it The x and y coordinates of all the 31 stops are obtained. The Table 4 1 gives the names of all 31 stops on the ring road network

Table 4.1 The major Stops on the Ring Road Network of Delhi

Stop Sequence No.	Name of the Stop
1	RAJA GARDEN
2	PANJABI BAGH
3	ROTHAK ROAD
4	SHAKURPUR
5	WAZIRPUR
6	PITAMPURA
7	SHALIMAR BAGH
8	AZADPUR
9	MODEL TOWN
10	GURUTEJ BAGH NAGAR
11	KHALSA COLLEGE
12	KHYBER PASS
13	OLD SECRETERIATE
14	I S B T
15	YAMUNA BAZAR
16	RED FORT
17	I P STADIUM
18	PRAGATI MAIDAN
19	NIZAMUDDIN
20	KALEKHA
21	BHAGAWAN NAGAR
22	ASHRAM
23	LAJPAT NAGAR
24	MOOL CHAND
25	SOUTH EXTENSION
26	A I.I M S
27	BHIKAJI
28	MOTI BAGH
29	DHAULA KUAN
30	BRAR SQUARE
31	MAYA PURI

The Table 4.2 gives the information of stops with respect to the desired probability distribution of passengers arrival at stops and its parameters In this Table EX refers to Exponential distribution , NR refers to Normal Distribution, GM refers to Gamma distribution, In Table 4 2, par_1, and par_2 are respectively first and second parameters of the distribution. For Exponential distribution only par_1 is considered as β, and there will be no par_2. For Normal distribution par_1 is considered as μ, the mean, and par_2 is considered as σ^2, the standard deviation. For

Gamma distribution par_1 is considered as α, the shape parameter, and par_2 is considered as β, the scale parameter.

Table 4.2 The Desired Probability Distributions of Passenger Arrivals at Stops

Stop Sequence No.	Desired Distribution	par_1 of the Distribution	par_2 of the Distribution
1	EX	2.5	
2	EX	3 2	
3	NR	2	0 5
4	EX	3	
5	GM	1.2	2
6	EX	3.6	
7	NR	3	0 5
8	EX	2.7	
9	EX	2 8	
10	EX	3 1	
11	NR	2 5	0.4
12	EX	3	
13	EX	3 5	
14	EX	2 9	
15	NR	4	1.33
16	EX	3 9	
17	EX	2 7	
18	GM	1.0	1 6
19	EX	3 1	
20	EX	2.3	
21	NR	2 4	.6
22	EX	2.9	
23	EX	2.87	
24	GM	2 1	1.3
25	EX	2.1	
26	EX	2 8	
27	NR	3	8
28	EX	2	
29	EX	2 95	
30	EX	3 8	
31	NR	2.7	.5

4.2.2 *FORMATION OF OD-MATRIX :*

From the complete daily demand matrix of size 1200×1200 of Delhi area first the daily demand matrix for all 71 nodes on the ring road network is extracted. The OD matrix for the 31 stops is then obtained by merging the OD of smaller nodes (the nodes with less demand and closely spaced) with that of pre-identified stops Thus obtained OD matrix of size 31×31 for all stops on the ring road network of Delhi area Now each element of this matrix represents the daily demand from one stop to the other, only on the ring road network. The table 4 3 gives the total productions of all the Stops in terms of daily demand

Table 4.3 Total Productions of all Stops

Stop Sequence No.	Total Productions of the Stop	Stop Sequence No.	Total Productions of the Stop
1	3560	16	716
2	4331	17	1593
3	1672	18	1645
4	2791	19	2141
5	1541	20	1129
6	1441	21	894
7	2580	22	487
8	3072	23	1045
9	2974	24	932
10	2613	25	1464
11	3573	26	837
12	3727	27	2260
13	3028	28	2781
14	383	29	1383
15	1339	30	1324
		31	1950

4.2.3 *IDENTIFICATION OF TERMINALS :*

After the stops on the ring road are identified, it is necessary to have few terminals which could supply buses to the ring road. A terminal in this model is defined a stop which supplies the buses to the route network under

consideration Each of the terminal will have some number of available buses, which is referred to as tot_bus in the model The stops which are not terminals will not have any buses , i e. tot_bus for such stop will be zero. With the initial time table given to the model as input, some number of buses will be scheduled from the each terminal Considering the remaining buses at each terminal, the model finds the best terminal from which the additional bus could be scheduled

Six terminals are identified out of 31 stops on the ring road based on the total demand from each stop to all the other stops It is assumed in this model that these terminals supply buses to the ring road network, i e all the buses originate from any of the terminals, move on the ring road serving all the intermediate stops and they finally stop at the respective terminals according to the pre-determined timetable. The Table 4 4 gives the names of the selected terminals on the ring road, and the number of buses available at theses terminals

Table 4.4 The Terminals on the Ring Road Network of Delhi.

Terminal No.	Name of the Terminal	No. of Buses at Terminal
1	RAJA GARDEN	8
8	AZADPUR	8
14	I.S.B.T	8
18	PRAGATI MAIDAN	8
23	LAJPAT NAGAR	8
29	DHAULA KUAN	8

4.2.4 *BUS INFORMATION :*

The simulation is initially started with 24 buses, four buses from each of the six terminals. Thus the FLEET_SIZE is equal to 24 before starting the simulation The maximum value of the FLEET_SIZE is MAX_BUS, the total number of buses in the system, which is equal to the 43 in this case study. This model suggests a few more additional buses which will be added to the FLEET_SIZE. The

headway of these buses is kept as 20 minutes at each terminal. The movement of buses on both directions (Clock Wise, and Anti Clock Wise) on the ring road is considered The Table 4 5 gives the information of buses initially under operation is given in terms of its starting terminal, starting time, and its direction of movement

Table 4.5 Information of Buses
(CW - Clock Wise, ACW - Anti Clock Wise)

Bus Sequence No.	Starting Terminal	Starting Time (Hrs.)	Direction of Movement
1	1	610	CW
2	1	630	ACW
3	1	650	CW
4	1	710	ACW
5	8	605	CW
6	8	625	ACW
7	8	645	CW
8	8	705	ACW
9	14	605	CW
10	14	625	ACW
11	14	645	CW
12	14	705	ACW
13	18	600	CW
14	18	620	ACW
15	18	640	CW
16	18	700	ACW
17	23	615	CW
18	23	635	ACW
19	23	655	CW
20	23	715	ACW
21	29	600	CW
22	29	620	ACW
23	29	640	CW
24	29	700	ACW

4.2.5 *PREPARING INITIAL TIMETABLE* :

An initial timetable is prepared to test the model. First the travel distance on each link is obtained from the data. For this case, ring road with 31 stops on it, the total number of links are 62, considering the movement of buses in both directions i.e clockwise, and anti clockwise. Assuming a constant speed of 15

kmph the travel time on each link are calculated in terms of minutes. Having given in the input the starting terminal and direction of movement of each of the 24 buses as shown in Table 4 5, the tentative timetable is prepared The timetable prepared to test this model covers the morning peak period from 0700 to 1000 hrs Considering the direction of movement of bus on ring road, and its starting terminal the next destination of that bus is obtained, and the departure time of the bus is computed using the following equation-

For a bus,

$$\text{departure time at stop}[i] = \text{departure time at stop}[i-1] + \text{travel time}(i-1, i) + \text{boarding/alighting time of passengers at stop}[i].$$

To develop initial timetable it is necessary to have the information of each link on the ring road network, in terms of travel distance, and travel time Table 4 6 gives the details of each link on the ring road of Delhi city

Table 4.6 Travel Distances, and Travel Times on Links of Ring Road.

From - Stop No.	To - Stop No.	Link Distance (meters)	Travel time (minutes)	Link Direction CW - Clock Wise ACW - Anti Clock Wise
1	2	2400	9	CW
2	1	2400	9	ACW
2	3	700	2	CW
3	2	700	2	ACW
3	4	1270	5	CW
4	3	1270	5	ACW
4	5	1380	5	CW
5	4	1380	5	ACW
5	6	780	3	CW
6	5	780	3	ACW
6	7	1050	4	CW

Table 4.6 cont'd

From - Stop No.	To - Stop No.	Link Distance (meters)	Travel time (minutes)	Link Direction CW - Clock Wise ACW - Anti Clock Wise
7	6	1050	4	ACW
7	8	860	3	CW
8	7	860	3	ACW
8	9	2230	8	CW
9	8	2230	8	ACW
9	10	1380	5	CW
10	9	1380	5	ACW
10	11	680	2	CW
11	10	680	2	ACW
11	12	1100	4	CW
12	11	1100	4	ACW
12	13	1150	4	CW
13	12	1150	4	ACW
13	14	1760	7	CW
14	13	1760	7	ACW
14	15	1230	4	CW
15	14	1230	4	ACW
15	16	880	3	CW
16	15	880	3	ACW
16	17	2000	8	CW
17	16	2000	8	ACW
17	18	1680	6	CW
18	17	1680	6	ACW
18	19	2880	11	CW
19	18	2880	11	ACW
19	20	1600	6	CW
20	19	1600	6	ACW
20	21	600	2	CW
21	20	600	2	ACW
21	22	1250	5	CW
22	21	1250	5	ACW
22	23	1600	6	CW
23	22	1600	6	ACW
23	24	1200	4	CW
24	23	1200	4	ACW
24	25	1620	6	CW
25	24	1620	6	ACW
25	26	970	3	CW
26	25	970	3	ACW
26	27	1870	7	CW
27	26	1870	7	ACW
27	28	1710	6	CW
28	27	1710	6	ACW

Table 4.6 *cont'd.*

From - Stop No.	To - Stop No.	Link Distance (meters)	Travel time (minutes)	Link Direction CW - Clock Wise ACW - Anti Clock Wise
28	29	2250	9	CW
29	28	2250	9	ACW
29	30	1590	6	CW
30	29	1590	6	ACW
30	31	3450	13	CW
31	30	3450	13	ACW
31	1	1510	6	CW
1	31	1510	6	ACW

4.2.6 STORING THE RELEVANT DATA IN INPUT FILES :

After preparing the input data it is required to store them in appropriate input files As mentioned in section 2 4 the input data must be stored in five input files For this case study the input files contain the following data -

1. Input file No. 1, 'STOP_XY.IN' contains the x and y coordinates, the desired distribution and its parameters, and the total number of buses available for each of the 31 stops on the ring road. The first element of this file is 31, the number of stops under consideration The maximum number of buses available in the system is equal to the sum of total number of buses of all stops This is taken as 48 in this case study (Refer Table 4 4)
2. The input file No. 2, 'BUS_INFO.IN' contains the starting terminal and direction of movement of each bus. The first element of this file is 50, which is the capacity of the bus taken in this model. In this file the information of 24 buses is stored, i e. the number of buses being operated on the ring road network before starting the simulation is 24. Thus the fleet size initially is equal to 24 The maximum value of the fleet size is the maximum number of buses in the system, i.e 48.
3. The input file No. 3, 'TIME_TAB.IN' contains the initial timetable of the public transit system developed for this model. The format of this file is first stop

number, then the number of departing bus, and its departure time The timetable generated as discussed in previous section is stored in this file For this case study, this file contains initial timetable for all the 24 buses serving the 31 stops on the ring road network

4 The input file No 4, 'LINKTIME IN' contains the travel time on each link of the ring road network. First the origin stop, then the destination stop, the travel time in minutes, and then the direction of that link is the format that is used in this file For this case study the travel times of all 62 links of the ring road are stored in this file

5 The input file No. 5, 'OD_MAT.IN' contains the daily demand matrix for all stops on the ring road network. This matrix for this case study is 31×31 symmetric matrix.

4.3 SIMULATION RUNS

The simulation process can be started by executing the program, i.e. by typing **'sim3000'** at DOS prompt. The main window will be displayed which will have four push buttons. *Esc* button - to stop the simulation and to exit to DOS; *Enter* button - to continue the simulation; *Ctrl+S* button - to display the information of all stops at any instant of time, and *Ctrl+B* button - to display the information of all buses at any instant of time The graphic display of this model is user interactive, and hence user can press any of the four buttons of his interest There will be a Time window which displays the time of simulation. The idea of how the model works is already discussed in the previous chapters.

Figure 4 1 shows display of Main window of this simulation model. The four push buttons *Esc* key, *Enter* key, *Ctrl+S* key, and *Ctrl+B* key can be clearly seen in the figure, which on activating performs different actions. The Time

window in the main screen displays the time 0535 Hrs. just before starting the simulation The Ring Road network of Delhi with 31 major stops on it can be observed All the vehicles, represented by green color icons are at their respective terminals as the simulation has just started, and their departure time has not yet reached.

Figure 4 2 shows the window which will be displayed when the model suggests an additional bus to be scheduled The two buttons to accept the additional bus, and to reject the additional bus can be activated by *Space bar* key, and *Esc* key respectively.

Figure 4 3 shows the Stop window, which will be displayed when the *Ctrl+S* key in the Main window is activated This window provides the sufficient information of all stops such as the number of passengers waiting at a stop, the average and total waiting time of passengers, the sequence number and departure time of next bus, ... etc.

Figure 4.4 shows the Bus window, which will be displayed when the *Ctrl+B* key in the Main window is activated. This window displays the information of all buses such as the number of passengers occupied the bus, the direction of movement of bus on the ring road, the sequence number of, and departure time at previous stop, and next stop

4.4 RESULTS AND DISCUSSION

The simulation for the case study of Delhi ring road network is carried over as discussed in section 4.3. The simulation process is started at 0600 hrs and terminated at 1130 hrs. thus covering the *morning peak period 0700 hrs to 1000 hrs, and the results obtained will* be discussed in the following sections

4.4.1 ADDITIONAL BUSES :

As stated in section 4.2.4, there are 24 buses in the system that are being operated initially. This model has simulated the public transit system on the ring road of Delhi city for the morning peak period and has suggested the 19 additional buses to ply on the ring road. The details of the additional buses scheduled on the ring road is given in the following table. These additional buses originate at their respective starting terminal, move along the ring road serving all the Stops on the ring road, and finally terminate at the then origin stop, i.e. starting terminal. The Table 4.6 gives the starting terminal, starting time, and the direction of movement of 19 additional buses.

Table 4.7 The Additional Buses Suggested by the Model

New Bus No.	Starting Terminal No.	Starting Time (Hrs.)	Direction of movement
25	29	610	CW
26	18	615	CW
27	14	620	CW
28	8	635	ACW
29	1	645	ACW
30	23	650	CW
31	14	705	ACW
32	18	720	CW
33	29	725	CW
34	8	730	ACW
35	23	745	ACW
36	1	750	ACW
37	8	810	CW

Table 4.7 Cont'd

New Bus No.	Starting Terminal No.	Starting Time Hrs.	Direction of movement
38	29	815	ACW
39	18	820	ACW
40	14	840	CW
41	18	845	CW
42	23	850	ACW
43	1	855	CW

4.4.2 *SAVING IN WAIT TIME*

In this section the save in the total waiting time of each stop is presented The model has suggested 19 new buses in addition to the 24 initial buses The additional buses are suggested based on the saving in total wait time, evaluating the best strategy as explained in section 2 5 5. Thus the scheduling of these 19 additional buses on the ring road has proved that the total waiting time at each stop is reduced It has been found that the total waiting time at a stop when all the additional buses, suggested by the model are accepted is less than the total waiting time at that stop when no additional bus is accepted This is presented in Table 4.7

Table 4.8 Total Saving in Passenger Wait Times

Stop No.	Total Wait Time Rejecting New Buses (Minutes)	Total Wait Time Accepting New Buses (Minutes)	Saving in Total Wait Time (Minutes)
1	2752	1932	820
2	2491	2281	210
3	3043	2530	513
4	3287	2382	905
5	2566	2369	197
6	3573	3109	464
7	3642	2963	679
8	2798	2141	657
9	2649	2544	105

Table 4.8 Cont'd

Stop No.	Total Wait Time Rejecting New Buses (Minutes)	Total Wait Time Accepting New Buses (Minutes)	Saving in Total Wait Time (Minutes)
10	2702	2180	522
11	3902	3377	525
12	2894	2397	497
13	3343	2976	367
14	3758	3294	464
15	2992	2472	520
16	3632	3157	475
17	3425	3051	374
18	4535	3929	606
19	2857	2317	540
20	3168	2496	672
21	2654	2183	471
22	2938	2585	353
23	2379	1902	477
24	3057	2461	596
25	2468	1836	632
26	3648	3172	476
27	3746	3282	464
28	4316	3442	874
29	5674	4261	1413
30	3258	2843	415
31	3494	2657	837

4.4.3 *NEW TIMETABLE*

The model has suggested 19 additional buses, and all these 19 additional buses were accepted to ply on the ring road The saving in total wait time due to the scheduling of additional buses is discussed in the previous section. The model has prepared the new timetable which includes the tentative timetable with 24 initial buses, and the timing of the 19 additional buses The buses with sequence numbers from 1 to 24 are the buses in the initial fleet, and with sequence numbers from 25 to 43 are the newly scheduled buses. The revised timetable is presented in Appendix.

CHAPTER V

CONCLUSIONS

5.1 CONCLUSIONS

This model has the capability for dynamic scheduling of vehicles on a major corridor of a metropolitan city. This model simulates the public transit system on any route network, and suggests the optimal utilization of vehicles which minimizes the passenger waiting times at all stops. Considering the large variations in passenger demand it is necessary to have a transit system with dynamic scheduling of vehicles for optimal design This model is flexible in terms of probability distributions of passenger arrival at stops. Passenger can arrive the stops as per any one of Exponential, Normal, and Gamma distributions, and the parameter of these distributions are as per user choice.

The model is developed with Object Oriented Programming (OOP) approach using C++ The model supports the user interactive graphics, so that the user can obtain the information of all stops and buses at any instant of time and/or quit the simulation at any instant of time The locations of all buses on the route network can be displayed at every 5 minute interval of time by the concurrent mode of animation The software is developed and run on Turbo C++ platform in an MS-DOS environment The size of the problem will be greatly influenced by the computational capabilities, and the available memory

This model is tested with a case study of Delhi city The ring road network of Delhi metropolitan area is considered, and the public transit on the ring road is simulated The data collected for a study 'Optimal Design of Urban Bus System for Delhi' is used partly as the input for this model A tentative timetable is prepared, which is fed to the model. The model has revised the timetable, which includes the scheduling of additional buses suggested by the model. For the case study, the model has suggested 19 new buses in addition to the 24 initial buses. The saving in wait time at all stops due to the schedule of these additional buses is also presented

5.2 *LIMITATIONS*

This model has some limitations which can be overcome by minor adjustments, and modifications to the software The following limitations are thought of

-

- The model can simulate the transit operations on a major corridor of city but not many corridors.
- The model can not take care of overlapping routes
- The arrival rate of passengers at a stop is assumed to be constant in this model Where as most of the passengers arrive the stops based on the scheduled bus timings.
- The model has not considered the stochastic nature of bus arrivals.
- This model can display the locations of buses on the route network at every five minute interval, but not at any intermediate time.

5.3 SCOPE FOR FUTURE WORK

Considering the limitations mentioned in the above section, there is a lot of scope for the future work on this area

- The model could be developed so as to make on-line decisions of optimal bus scheduling for a city route network.
- The simulation for dynamic scheduling is limited to a single corridor in this model, which can be extended to many corridors with overlapping routes.

APPENDIX

Timetable revised by the model.

Includes the schedules of additional buses.

1 Stop No.	2 Bus No.	3 Departure Time (Hrs)	1 Stop No	2 Bus No.	3 Departure Time (Hrs)	1 Stop No.	2 Bus No.	3 Departure Time (Hrs)
1	1	610	1	5	826	2	28	657
1	21	625	1	18	827	2	3	659
1	2	630	1	11	836	2	23	714
1	25	635	1	22	847	2	10	717
1	29	645	1	32	850	2	17	724
1	3	650	1	43	855	2	8	727
1	6	656	1	1	902	2	14	733
1	23	705	1	7	906	2	13	739
1	28	706	1	20	907	2	34	752
1	4	710	1	2	922	2	26	754
1	17	715	1	24	927	2	12	757
1	10	726	1	29	937	2	31	757
1	13	730	1	35	937	2	30	759
1	8	736	1	3	942	2	33	759
1	14	742	1	39	942	2	19	804
1	26	745	1	4	1002	2	9	805
1	30	750	1	41	1015	2	16	813
1	33	750	1	37	1031	2	18	818
1	36	750	1	40	1031	2	15	819
1	19	755	1	36	1042	2	27	820
1	9	756	1	38	1042	2	5	835
1	34	801	1	42	1042	2	22	838
1	12	806	1	43	1147	2	11	845
1	31	806	2	1	619	2	20	858
1	15	810	2	21	634	2	32	859
1	27	811	2	25	644	2	43	904
1	16	822	2	6	647	2	2	913

1	2	3	1	2	3	1	2	3
2	7	915	3	2	911	4	32	906
2	24	918	3	24	916	4	24	911
2	29	928	3	7	917	4	43	911
2	35	928	3	29	926	4	29	921
2	39	933	3	35	926	4	35	921
2	4	953	3	39	931	4	7	922
2	41	1024	3	4	951	4	39	926
2	36	1033	3	41	1026	4	4	946
2	38	1033	3	36	1031	4	36	1026
2	42	1033	3	38	1031	4	38	1026
2	37	1040	3	42	1031	4	42	1026
2	40	1040	3	37	1042	4	41	1031
3	1	621	3	40	1042	4	37	1047
3	21	636	4	1	626	4	40	1047
3	6	645	4	6	640	5	1	631
3	25	646	4	21	641	5	6	635
3	28	655	4	28	650	5	28	645
3	3	701	4	25	651	5	21	646
3	10	715	4	3	706	5	25	656
3	23	716	4	10	710	5	10	705
3	8	725	4	8	720	5	3	711
3	17	726	4	23	721	5	8	715
3	14	731	4	14	726	5	14	721
3	13	741	4	17	731	5	23	726
3	34	750	4	34	745	5	17	736
3	12	755	4	13	746	5	34	740
3	31	755	4	12	750	5	12	745
3	26	756	4	31	750	5	31	745
3	30	801	4	26	801	5	13	751
3	33	801	4	16	806	5	16	801
3	19	806	4	30	806	5	18	806
3	9	807	4	33	806	5	26	806
3	16	811	4	18	811	5	30	811
3	18	816	4	19	811	5	33	811
3	15	821	4	9	812	5	19	816
3	27	822	4	15	826	5	9	817
3	22	836	4	27	827	5	22	826
3	5	837	4	22	831	5	15	831
3	11	847	4	5	842	5	27	832
3	20	856	4	20	851	5	20	846
3	32	901	4	11	852	5	5	847
3	43	906	4	2	906	5	11	857

1	2	3	1	2	3	1	2	3
5	2	901	6	2	858	7	2	854
5	24	906	6	11	900	7	5	854
5	32	911	6	24	903	7	24	859
5	29	916	6	29	913	7	11	904
5	35	916	6	35	913	7	29	909
5	43	916	6	32	914	7	35	909
5	39	921	6	39	918	7	39	914
5	7	927	6	43	919	7	32	918
5	4	941	6	7	930	7	43	923
5	36	1021	6	4	938	7	4	934
5	38	1021	6	36	1018	7	7	934
5	42	1021	6	38	1018	7	36	1014
5	41	1036	6	42	1018	7	38	1014
5	37	1052	6	41	1039	7	42	1014
5	40	1052	6	37	1055	7	41	1043
6	6	632	6	40	1055	7	37	1059
6	1	634	7	6	628	7	40	1059
6	28	642	7	1	638	8	5	605
6	21	649	7	28	638	8	6	625
6	25	659	7	21	653	8	28	635
6	10	702	7	10	658	8	1	641
6	8	712	7	25	703	8	7	645
6	3	714	7	8	708	8	10	655
6	14	718	7	14	714	8	21	656
6	23	729	7	3	718	8	8	705
6	34	737	7	23	733	8	25	706
6	17	739	7	34	733	8	14	711
6	12	742	7	12	738	8	3	721
6	31	742	7	31	738	8	34	730
6	13	754	7	17	743	8	12	735
6	16	758	7	16	754	8	31	735
6	18	803	7	13	758	8	23	736
6	26	809	7	18	759	8	17	746
6	30	814	7	26	813	8	16	751
6	33	814	7	30	818	8	18	756
6	19	819	7	33	818	8	13	801
6	9	820	7	22	819	8	37	810
6	22	823	7	19	823	8	22	816
6	15	834	7	9	824	8	26	816
6	27	835	7	15	838	8	30	821
6	20	843	7	20	839	8	33	821
6	5	850	7	27	839	8	19	826

1	2	3	1	2	3	1	2	3
8	9	827	9	26	824	10	20	823
8	20	836	9	20	828	10	37	823
8	15	841	9	30	829	10	26	829
8	27	842	9	33	829	10	30	834
8	2	851	9	19	834	10	33	834
8	24	856	9	9	835	10	2	838
8	5	857	9	2	843	10	19	839
8	29	906	9	24	848	10	9	840
8	35	906	9	15	849	10	24	843
8	11	907	9	27	850	10	29	853
8	39	911	9	29	858	10	35	853
8	6	917	9	35	858	10	15	854
8	32	921	9	39	903	10	27	855
8	43	926	9	6	909	10	39	858
8	28	927	9	11	915	10	6	904
8	4	931	9	28	919	10	28	914
8	7	937	9	4	923	10	4	918
8	8	957	9	32	929	10	11	920
8	36	1011	9	43	934	10	32	934
8	38	1011	9	8	949	10	43	939
8	42	1011	9	36	1003	10	8	944
8	34	1022	9	38	1003	10	36	958
8	41	1046	9	42	1003	10	38	958
8	37	1102	9	34	1014	10	42	958
8	40	1102	9	41	1054	10	34	1009
9	5	613	9	40	1110	10	41	1059
9	10	647	10	5	618	10	40	1115
9	1	649	10	10	642	11	5	620
9	7	653	10	1	654	11	10	640
9	14	703	10	7	658	11	1	656
9	21	704	10	14	658	11	14	656
9	25	714	10	21	709	11	7	700
9	12	727	10	25	719	11	21	711
9	31	727	10	12	722	11	12	720
9	3	729	10	31	722	11	31	720
9	16	743	10	3	734	11	25	721
9	23	744	10	16	738	11	3	736
9	18	748	10	18	743	11	16	736
9	17	754	10	23	749	11	18	741
9	22	808	10	17	759	11	23	751
9	13	809	10	22	803	11	17	801
9	37	818	10	13	814	11	22	801

1	2	3	1	2	3	1	2	3
11	13	816	12	20	817	13	20	813
11	20	821	12	13	820	13	13	824
11	37	825	12	37	829	13	2	828
11	26	831	12	2	832	13	24	833
11	2	836	12	26	835	13	37	833
11	30	836	12	24	837	13	26	839
11	33	836	12	30	840	13	29	843
11	19	841	12	33	840	13	35	843
11	24	841	12	19	845	13	30	844
11	9	842	12	9	846	13	33	844
11	29	851	12	29	847	13	39	848
11	35	851	12	35	847	13	19	849
11	15	856	12	39	852	13	9	850
11	39	856	12	6	858	13	6	854
11	27	857	12	15	900	13	15	904
11	6	902	12	27	901	13	28	904
11	28	912	12	28	908	13	27	905
11	4	916	12	4	912	13	4	908
11	11	922	12	11	926	13	11	930
11	32	936	12	8	938	13	8	934
11	43	941	12	32	940	13	32	944
11	8	942	12	43	945	13	36	948
11	36	956	12	36	952	13	38	948
11	38	956	12	38	952	13	42	948
11	42	956	12	42	952	13	43	949
11	34	1007	12	34	1003	13	34	959
11	41	1101	12	41	1105	13	41	1109
11	40	1117	12	40	1121	13	40	1125
12	5	624	13	5	628	14	9	605
12	10	636	13	10	632	14	27	620
12	14	652	13	14	648	14	10	625
12	1	700	13	1	704	14	5	635
12	7	704	13	7	708	14	14	641
12	21	715	13	12	712	14	11	645
12	12	716	13	31	712	14	12	705
12	31	716	13	21	719	14	31	705
12	25	725	13	16	728	14	1	711
12	16	732	13	25	729	14	7	715
12	18	737	13	18	733	14	16	721
12	3	740	13	3	744	14	18	726
12	23	755	13	22	753	14	21	726
12	22	757	13	23	759	14	25	736
12	17	805	13	17	809	14	22	746

1	2	3	1	2	3	1	2	3
14	3	751	15	18	722	16	7	722
14	20	806	15	21	730	16	21	733
14	23	806	15	25	740	16	22	739
14	17	816	15	22	742	16	25	743
14	2	821	15	3	755	16	3	758
14	24	826	15	20	802	16	20	759
14	13	831	15	23	810	16	23	813
14	29	836	15	2	817	16	2	814
14	35	836	15	17	820	16	24	819
14	37	840	15	24	822	16	17	823
14	40	840	15	29	832	16	29	829
14	39	841	15	35	832	16	35	829
14	26	846	15	13	835	16	39	834
14	6	847	15	39	837	16	13	838
14	30	851	15	6	843	16	6	840
14	33	851	15	37	844	16	37	847
14	19	856	15	40	844	16	40	847
14	9	857	15	26	850	16	28	850
14	28	857	15	28	853	16	26	853
14	4	901	15	30	855	16	4	854
14	15	911	15	33	855	16	30	858
14	27	912	15	4	857	16	33	858
14	10	917	15	19	900	16	19	903
14	8	927	15	10	913	16	10	910
14	11	937	15	15	915	16	15	918
14	36	941	15	8	923	16	8	920
14	38	941	15	36	937	16	36	934
14	42	941	15	38	937	16	38	934
14	32	951	15	42	937	16	42	934
14	34	952	15	34	948	16	34	945
14	43	956	15	12	953	16	12	950
14	12	957	15	31	953	16	31	950
14	31	957	15	32	955	16	32	958
14	41	1116	15	43	1000	16	43	1003
14	40	1132	15	41	1120	16	41	1123
15	9	609	16	9	612	17	9	620
15	27	624	16	27	627	17	14	626
15	14	637	16	14	634	17	27	635
15	5	639	16	5	642	17	5	650
15	11	649	16	11	652	17	11	700
15	1	715	16	16	714	17	16	706
15	16	717	16	1	718	17	18	711
15	7	719	16	18	719	17	1	726

1	2	3	1	2	3	1	2	3
17	7	730	18	16	700	18	41	1137
17	22	731	18	11	706	19	26	626
17	21	741	18	32	720	19	9	637
17	20	751	18	22	725	19	15	651
17	25	751	18	1	732	19	27	652
17	2	806	18	7	736	19	18	654
17	3	806	18	20	745	19	5	707
17	24	811	18	21	747	19	22	714
17	23	821	18	25	757	19	11	717
17	29	821	18	2	800	19	32	731
17	35	821	18	24	805	19	20	734
17	39	826	18	3	812	19	1	743
17	17	831	18	29	815	19	7	747
17	6	832	18	35	815	19	2	749
17	28	842	18	39	820	19	24	754
17	4	846	18	6	826	19	21	758
17	13	846	18	23	827	19	29	804
17	37	855	18	28	836	19	35	804
17	40	855	18	17	837	19	25	808
17	26	901	18	4	840	19	6	815
17	10	902	18	41	845	19	3	823
17	30	906	18	13	852	19	28	825
17	33	906	18	10	856	19	4	829
17	19	911	18	37	901	19	23	838
17	8	912	18	40	901	19	10	845
17	15	926	18	8	906	19	17	848
17	36	926	18	26	907	19	8	855
17	38	926	18	14	912	19	41	856
17	42	926	18	30	912	19	14	901
17	34	937	18	33	912	19	36	909
17	12	942	18	19	917	19	38	909
17	31	942	18	36	920	19	42	909
17	32	1006	18	38	920	19	37	912
17	43	1011	18	42	920	19	40	912
17	41	1131	18	34	931	19	34	920
18	13	600	18	15	932	19	30	923
18	26	615	18	12	936	19	33	923
18	14	620	18	31	936	19	12	925
18	9	626	18	16	952	19	31	925
18	15	640	18	32	1012	19	19	928
18	27	641	18	43	1017	19	16	941
18	5	656	18	39	1112	19	43	1028

1	2	3	1	2	3	1	2	3
19	39	1101	20	39	1055	21	43	1036
20	26	632	21	13	619	21	39	1053
20	9	643	21	26	634	22	13	624
20	18	648	21	9	645	22	26	639
20	15	657	21	18	646	22	18	641
20	27	658	21	15	659	22	9	650
20	22	708	21	27	700	22	22	701
20	5	713	21	22	706	22	15	704
20	11	723	21	5	715	22	27	705
20	20	728	21	11	725	22	5	720
20	32	737	21	20	726	22	20	721
20	2	743	21	32	739	22	11	730
20	24	748	21	2	741	22	2	736
20	1	749	21	24	746	22	24	741
20	7	753	21	1	751	22	32	744
20	29	758	21	7	755	22	29	751
20	35	758	21	29	756	22	35	751
20	21	804	21	35	756	22	1	756
20	6	809	21	21	806	22	7	800
20	25	814	21	6	807	22	6	802
20	28	819	21	25	816	22	21	811
20	4	823	21	28	817	22	28	812
20	3	829	21	4	821	22	4	816
20	10	839	21	3	831	22	25	821
20	23	844	21	10	837	22	10	832
20	8	849	21	23	846	22	3	836
20	17	854	21	8	847	22	8	842
20	14	855	21	14	853	22	14	848
20	41	902	21	17	856	22	23	851
20	36	903	21	36	901	22	36	856
20	38	903	21	38	901	22	38	856
20	42	903	21	42	901	22	42	856
20	34	914	21	41	904	22	17	901
20	37	918	21	34	912	22	34	907
20	40	918	21	12	917	22	41	909
20	12	919	21	31	917	22	12	912
20	31	919	21	37	920	22	31	912
20	30	929	21	40	920	22	37	925
20	33	929	21	30	931	22	40	925
20	19	934	21	33	931	22	16	928
20	16	935	21	16	933	22	30	936
20	43	1034	21	19	936	22	33	936

1	2	3	1	2	3	1	2	3
22	19	941	23	18	927	24	31	902
22	43	1041	23	37	931	24	16	918
22	39	1048	23	40	931	24	41	919
23	17	615	23	30	942	24	18	923
23	13	630	23	33	942	24	37	935
23	18	635	23	19	947	24	40	935
23	26	645	23	20	1007	24	33	946
23	30	650	23	35	1037	24	20	1003
23	19	655	23	39	1042	24	35	1033
23	22	655	23	43	1047	24	39	1038
23	9	656	23	42	1142	24	43	1051
23	15	710	24	17	619	24	42	1138
23	27	711	24	13	634	25	17	625
23	20	715	24	26	649	25	13	640
23	5	726	24	22	651	25	22	645
23	2	730	24	30	654	25	26	655
23	24	735	24	19	659	25	30	700
23	11	736	24	9	700	25	19	705
23	29	745	24	15	714	25	9	706
23	35	745	24	27	715	25	2	720
23	32	750	24	2	726	25	15	720
23	6	756	24	5	730	25	27	721
23	1	802	24	24	731	25	24	725
23	7	806	24	11	740	25	29	735
23	28	806	24	29	741	25	5	736
23	4	810	24	6	752	25	6	746
23	21	817	24	32	754	25	11	746
23	10	826	24	28	802	25	28	756
23	25	827	24	1	806	25	4	800
23	8	836	24	4	806	25	32	800
23	3	842	24	7	810	25	1	812
23	14	842	24	21	821	25	7	816
23	36	850	24	10	822	25	10	816
23	38	850	24	25	831	25	8	826
23	42	850	24	8	832	25	21	827
23	23	857	24	14	838	25	14	832
23	34	901	24	3	846	25	25	837
23	12	906	24	36	846	25	36	840
23	31	906	24	38	846	25	38	840
23	17	907	24	34	857	25	34	851
23	41	915	24	23	901	25	3	852
23	16	922	24	12	902	25	12	856

1	2	3	1	2	3	1	2	3
25	31	856	26	3	855	27	25	847
25	16	912	26	16	909	27	3	902
25	18	917	26	23	910	27	16	902
25	41	925	26	18	914	27	18	907
25	37	941	26	41	928	27	23	917
25	40	941	26	37	944	27	41	935
25	33	952	26	40	944	27	20	947
25	20	957	26	20	954	27	37	951
25	35	1027	26	33	955	27	40	951
25	39	1032	26	35	1024	27	33	1002
25	43	1057	26	39	1029	27	35	1017
25	42	1132	26	43	1100	27	39	1022
26	17	628	26	42	1129	27	43	1107
26	22	642	27	17	635	27	42	1122
26	13	643	27	22	635	28	22	629
26	26	658	27	13	650	28	17	641
26	30	703	27	26	705	28	13	656
26	19	708	27	2	710	28	2	704
26	9	709	27	30	710	28	24	709
26	2	717	27	19	715	28	26	711
26	24	722	27	24	715	28	30	716
26	15	723	27	9	716	28	29	719
26	27	724	27	29	725	28	19	721
26	29	732	27	15	730	28	9	722
26	5	739	27	27	731	28	6	730
26	6	743	27	6	736	28	15	736
26	11	749	27	5	746	28	27	737
26	28	753	27	28	746	28	28	740
26	4	757	27	4	750	28	4	744
26	32	803	27	11	756	28	5	752
26	10	813	27	10	806	28	10	800
26	1	815	27	32	810	28	11	802
26	7	819	27	8	816	28	8	810
26	8	823	27	1	822	28	14	816
26	14	829	27	14	822	28	32	816
26	21	830	27	7	826	28	36	824
26	36	837	27	36	830	28	38	824
26	38	837	27	38	830	28	1	828
26	25	840	27	21	837	28	7	832
26	34	848	27	34	841	28	34	835
26	12	853	27	12	846	28	12	840
26	31	853	27	31	846	28	31	840

1	2	3	1	2	3	1	2	3
28	21	843	29	34	826	30	14	801
28	25	853	29	12	831	30	5	807
28	16	856	29	31	831	30	36	809
28	18	901	29	1	837	30	11	817
28	3	908	29	7	841	30	34	820
28	23	923	29	16	847	30	12	825
28	20	941	29	18	852	30	31	825
28	41	941	29	21	852	30	32	831
28	37	957	29	25	902	30	16	841
28	40	957	29	22	912	30	1	843
28	33	1008	29	3	917	30	18	846
28	35	1011	29	20	932	30	7	847
28	39	1016	29	23	932	30	22	906
28	43	1113	29	41	950	30	3	923
28	42	1116	29	24	952	30	20	926
29	21	600	29	35	1002	30	24	946
29	25	610	29	37	1006	30	35	956
29	22	620	29	40	1006	30	41	956
29	23	640	29	39	1007	30	39	1001
29	17	650	29	33	1017	30	37	1012
29	2	655	29	38	1107	30	40	1012
29	24	700	29	42	1107	30	38	1101
29	13	705	29	43	1122	30	42	1101
29	29	710	30	21	606	30	43	1128
29	26	720	30	25	616	31	21	619
29	6	721	30	23	646	31	25	629
29	30	725	30	2	649	31	2	636
29	33	725	30	17	656	31	29	651
29	19	730	30	29	704	31	23	659
29	9	731	30	13	711	31	6	702
29	28	731	30	6	715	31	17	709
29	4	735	30	28	725	31	28	712
29	15	745	30	26	726	31	4	716
29	27	746	30	4	729	31	13	724
29	10	751	30	30	731	31	10	732
29	5	801	30	33	731	31	26	739
29	8	801	30	19	736	31	8	742
29	14	807	30	9	737	31	30	744
29	11	811	30	10	745	31	33	744
29	36	815	30	15	751	31	14	748
29	38	815	30	27	752	31	19	749
29	32	825	30	8	755	31	9	750

1	2	3
31	36	756
31	15	804
31	27	805
31	34	807
31	12	812
31	31	812
31	5	820
31	16	828
31	11	830
31	18	833
31	32	844
31	22	853
31	1	856
31	7	900
31	20	913
31	24	933
31	3	936
31	35	943
31	39	948
31	41	1009
31	37	1025
31	40	1025
31	38	1048
31	42	1048
31	43	1141

REFERENCES

1. Andersson P.-A , Scalia-Tomba G.-A (1981) "A Mathematical Model of an Urban Bus Route." *Transpn Res* **15B**, 249-266.
2. Andersson P -A., Hermansson A , Tengvald E and Scalia-Tomba G -P (1979) "Analysis and Simulation of an Urban Bus Route " *Transpn Res* **13A**, 439-466.
3. Ceder A. (1986) "Methods for Creating Bus Timetables " *Transpn Res* **21A**, 59-83
4. Dhingra S L (1980), "Simulation of Routing and Scheduling of City Bus Transit Network " *A Ph D Thesis* IIT Kanpur.
5. Dumas Y , Soumis F and Desrosiers J (1990) "Optimizing the Schedule for a Fixed Vehicle Path with Convex Inconvienience Costs." *Transpn Sci* **24**, 145-152
6. Frieman M (1975) "Rapid Calculation of a certain Enumeration Problem Encountered in a Public Transportation Network." *Transpn Res* **9**, 203-204.
7. Friedman M. (1974) "A mathematial Programming Model for Optimal Scheduling of Buses' Departures under Deterministic Conditions." *Transpn Res* **10**, 83-90
8. Lafore R. (1994) "Object Oriented Programming in Turbo C++." *The Waite Groups*, Galgotia Publications Pvt. Ltd., India.
9. Law A.M., Kelton W D. (1991) "Simulation Modeling and Analysis " *2nd ed.* Mc Graw-Hill, Inc.
10. RITES. (1993) "Optimal Design of Urban Bus System for Delhi " *Final Report*, RITES.
11. Scheele S. (1980) "A Supply Model for Public Transit Services." *Transpn Res.* **14B**, 133-146.

A 123574

Date Due

CE-1997-M-KUM-SIM